

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра автоматики та управління в технічних системах**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра  
за освітньо-професійною програмою «Програмне забезпечення  
інформаційно-комунікаційних систем»  
спеціальності 121 «Інженерія програмного забезпечення»  
на тему: «Внутрішньобанківська інформаційна система локального  
управління рахунками на основі WPF-технології»**

Виконав:

студент IV курсу, групи ІТ-61

Бондаренко Олександр Леонідович

\_\_\_\_\_

Керівник:

професор каф. АУТС, д.е.н.,

Шемаєв Володимир Миколайович

\_\_\_\_\_

Рецензент:

С.н.с. ІТГІП НАН України, д.т.н.,

Биченок Миколо Миколайович

\_\_\_\_\_

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент (-ка) \_\_\_\_\_

Київ — 2020 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматики та управління в технічних системах**

Рівень вищої освіти — перший (бакалаврський)

Спеціальність — 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Бондаренку Олександру Леонідовичу**

1. Тема проєкту «Внутрішньобанківська інформаційна система локального управління рахунками на основі WPF-технології», керівник проєкту Шемаєв Володимир Миколайович, професор кафедри АУТС, затверджені наказом по університету від «7»травня 2020 р. № 1081-с
2. Термін подання студентом проєкту: 9.06.2020
3. Вихідні дані до проєкту: мова програмування C#, середовище розробки Microsoft Visual Studio, технологія WPF, система управління базами даних MS SQL Server.
4. Зміст пояснювальної записки: опис та аналіз об'єкту дослідження, моделювання та конструювання системи, опис програми, тестування, впровадження програмного забезпечення.
5. Перелік графічного матеріалу: діаграма прецедентів, структурна схема інтерфейсу додатку, діаграма класів, структура бази даних.

6. Дата видачі завдання 3 березня 2020 року

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	10.04.2020	Виконано
2.	Аналіз теоретичних матеріалів та вивчення предметної області	13.04.2020	Виконано
3.	Аналіз існуючих рішень	20.04.2020	Виконано
4.	Виготовлення макету інформаційної системи	24.04.2020	Виконано
5.	Вибір методів та засобів реалізації	26.04.2020	Виконано
6.	Розробка структури БД та початок проєктування системи	29.04.2020	Виконано
7.	Завершення реалізації проєкту	14.05.2020	Виконано
8.	Виконання графічних документів	17.05.2020	Виконано
9.	Оформлення пояснювальної записки	18.05.2020	Виконано
10.	Подання ДП на попередній захист	25.05.2020	Виконано
11.	Подання ДП рецензенту	08.06.2020	Виконано
12.	Подання ДП на основний захист	09.06.2020	Виконано

Студент

Олександр БОНДАРЕНКО

Керівник

Володимир ШЕМАЄВ

## АНОТАЦІЯ

Бондаренко О.Л. Внутрішньобанківська інформаційна система локального управління рахунками на основі WPF технології. КІП ім. Ігоря Сікорського, Київ, 2020.

Проект містить 5 розділів, 72 сторінки, 23 рисунка, 22 таблиці, 15 посилань на літературні джерела та 4 конструкторських документів.

Ключові слова: банк, банкінг, фінансова операція, транзакція, автентифікація, клієнт банку, адміністратор, менеджер, C#, WPF, база даних.

Об'єктом розробки є система управління рахунками банку.

Мета розробки — створити додаток зі зручним інтерфейсом, в якому можна буде реєструвати власний банківський рахунок та здійснювати фінансові операції.

У дипломному проєкті розроблено систему, яка зберігає дані банківської системи, дає змогу реєструвати новий банк в системі, нових користувачів та рахунки для них, надає змогу здійснювати фінансові операції з цими рахунками. Було проведено аналіз програм-аналогів, що дозволило прийняти ряд рішень щодо функціоналу програмного забезпечення. Було обрано технологію WPF, яка дозволить розробити зручний, лаконічний та гнучкий інтерфейс для приємного користування додатком. Значну увагу було приділено базі даних, що має зберігати та редагувати інформацію про усю банківську систему.

Отримані результати можуть бути корисні для банків, що не мають програмного забезпечення. Функціонал системи буде розширюватись да вдосконалюватись.

## **SUMMARY**

Bondarenko O.L. Intrabank information system of local account management based on WPF technology. Igor Sikorsky Kyiv Polytechnic Institute , Kyiv, 2020.

The project contains 5 sections, 72 pages, 23 figures, 22 schemes, 15 references to literature sources and 4 design documents.

Keywords: bank, banking, financial transaction, transaction, authentication, bank client, administrator, manager, C #, WPF, database.

The object of development is a bank account management system. The purpose of the development is to create an application with a user-friendly interface where you can register your own bank account and perform financial transactions. The diploma project contains a system that stores data from the banking system, allows you to register a new bank in the system, new users and accounts for them, allows you to perform financial transactions with these accounts. An analysis of analog programs was performed, which allowed to make a number of decisions about the functionality of the software. WPF technology was chosen that will allow you to develop a user-friendly, concise and flexible interface for pleasant use of the application. Considerable attention was paid to the database, which must store and edit information about the entire banking system.

The obtained results may be useful for banks that do not have software. The functionality of the system will be expanded and improved.

Поз.	Формат	Позначення	Найменування	Кількість аркушів	№ прим.	Примітки
1			Документація загальна			
2						
3			Знову розроблена			
4						
5	A4	IT61.050БАК.004 ПЗ	Внутрішньобанківська	72		
6			інформаційна система			
7			локального управління			
8			рахунками на основі			
9			WPF-технології.			
10			Пояснювальна записка.			
11	A3	IT61.050БАК.004 Д1	Внутрішньобанківська	1		
12			інформаційна система			
13			локального управління			
14			рахунками на основі			
15			WPF-технології.			
16			Діаграма прецедентів.			
17	A3	IT61.050БАК.004 Д2	Внутрішньобанківська	1		
18			інформаційна система			
19			локального управління			
20			рахунками на основі			
21			WPF-технології.			
22			Структурна схема			
23			інтерфейсу додатку.			
24	A3	IT61.050БАК.004 Д3	Внутрішньобанківська	1		
25			інформаційна система			
26			локального управління			
27			рахунками на основі			
28			WPF-технології.			
29			Діаграма класів.			
30	A3	IT61.050БАК.004 Д4	Внутрішньобанківська	1		
31			інформаційна система			
32			локального управління			
33			рахунками на основі			
34			WPF-технології.			
35			Структура бази даних.			
36						
37						
38						
			IT61.050БАК.004 ТП			
Змн.	Лист	№ докум.	Підпис	Дата		
Розроб.		Бондаренко О.Л.			Лім.	Лист
Перевір.		Шемаєв В.М.				Листів
Н. Контр.					КПІ ім. Ігоря Сікорського	
Затверд.					кафедра АУТС гр. ІТ-61	

**Пояснювальна записка  
до дипломного проєкту  
на тему: « Внутрішньобанківська інформаційна  
система локального управління рахунками  
на основі WPF-технології »**

Київ — 2020 року

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	4
ВСТУП.....	5
1 ОПИС ТА АНАЛІЗ ОБ’ЄКТА ДОСЛІДЖЕННЯ.....	9
1.1 Загальні відомості .....	9
1.2 Дослідження та статистика банківських додатків в Україні .....	10
1.2.1 Дослідження Мінфіну .....	12
1.2.3 Аналіз досліджень .....	14
1.3 Власне дослідження та збір статистики.....	15
1.4 Розвиток в умовах пандемії COVID-19 .....	19
1.5 Огляд аналогів .....	20
1.5.1 Аналіз додатку «Privat24» .....	20
1.5.2 Аналіз додатку «Альфа-Банк» .....	23
1.5.3 Порівняння .....	24
Висновки до розділу .....	25
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ СИСТЕМИ .....	26
2.1 Вибір функціоналу системи .....	26
2.2 Розробка деяких алгоритмів.....	29
2.3 Проєктування бази даних для системи .....	30
2.4 Вибір інструментів.....	34
2.4.1 Вибір IDE .....	34
2.4.2 Основні рішення щодо збереження інформації .....	36

					IT61.050БАК.004 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розроб.		Бондаренко О.Л.			Внутрішньобанківська інформаційна система локального управління рахунками на основі WPF-технології. Пояснювальна записка	Лім.	Лист	Листів
Перевір.		Шемасв В.М.					2	72
Н. Контр.								
Затверд.						КПІ ім. Ігоря Сікорського кафедра АУТС гр. ІТ-61		



2.4.3 Вибір інструменту для створення графічного інтерфейсу .....	38
Висновки до розділу .....	39
3 ОПИС ПРОГРАМИ.....	40
3.1 Реалізація системи ролей.....	40
3.2 Основні класи системи .....	40
3.3 Реалізація бази даних.....	42
3.4 Структура вікон додатку .....	47
Висновки до розділу .....	49
4 ТЕСТУВАННЯ ПРОГРАМИ .....	51
4.1 Загальні відомості .....	51
4.2 Модульне тестування.....	52
4.3 Тестування сценаріїв .....	55
4.4 Тестування навантаженням.....	58
Висновки до розділу .....	59
5 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	60
5.1 Підготовка до запуску програми .....	60
5.1.1 Вимоги до програмного та апаратного забезпечення .....	60
5.1.2 Вимоги для запуску.....	60
5.1.3 Підключення до MS SQL Server .....	61
5.2 Інструкція з використання клієнту банку .....	61
5.3 Інструкція з використання адміністратора.....	66
Висновки до розділу .....	69
ВИСНОВКИ.....	70
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	72

					ІТ61.050БАК.004 ПЗ	Лист
						3
Змн.	Лист	№ докум.	Підпис	Дата		

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

Банк — юридична особа, яка на підставі банківської ліцензії має виключене право надати банківська послуги. Учасниками банку можуть бути юридичні та фізичні особи, резиденти та нерезиденти, а також держава в особі Кабінету Міністрів України або уповноважені ним органи. [1]

Банкінг, електронний банкінг — це діяльність банку з надання комплексу послуг клієнтам за допомогою комп'ютерних технологій. [2]

Інтерфейс — засіб зручної взаємодії користувача з інформаційною системою. Сукупність засобів для обробки та відбиття інформації, якнайбільше пристосованих для зручності користувача. [3]

БД (база даних) — це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів.[4]

IDE (інтегроване середовище розробки) — це програмне забезпечення, яке надає всі засоби для розробника, необхідні в процесі розробки. [5]

СУБД (система управління базами даних) — це програмне забезпечення для створення та роботи з базами даних. [6]

WPF (Windows Presentation Foundation) — є частина екосистеми платформи .NET і являє собою підсистему для побудови графічних інтерфейсів. [7]

Рефакторинг — це контрольований процес поліпшення коду, без написання нової функціональності. [8]

Уніфікація — найпоширеніший і ефективний метод стандартизації, який передбачає приведення об'єктів до однотипності на основі встановлення раціонального числа їх різновидів. [9]

Фреймворк — це певний набір бібліотек, який полегшує розробку будь-яких продуктів: web-сайтів і web-сервісів, мобільних або десктопних додатків. [10]

					ІТ61.050БАК.004 ПЗ	Лист
						4
Змн.	Лист	№ докум.	Підпис	Дата		

## ВСТУП

Фінансова система упродовж усієї історії людства займає значну роль в житті людини. Споконвіку люди передають та обмінюються товарами та послугами в будь-яких галузях. У Вавилоні протягом вісімнадцятого століття до нашої ери вже відбулися обміни між купцями та священиками, подібно до того, як у Римі були також економічні обміни, але більше між людьми, ніж між установами. Однак лише до епохи Відродження виникли банки, як ми їх знаємо сьогодні. Банк означає «письмовий стіл» італійською мовою і натякав на столи, де сиділи перші флорентійські євреї, які працювали банком. Родини банкірів почали існувати, а також власні банківські центри, де здійснювалися переміщення срібних та золотих монет. Все це надзвичайно зросло протягом 18-19 століть, і це було у 20 столітті, коли розпочалося регулювання фінансів практично на всій планеті, намагаючись встановити правила та основи для банківської діяльності, як ми це знаємо сьогодні.

Гроші розповсюджуються, їх стає більше, з'являється безліч нових установ, що контролюють та полегшують обіг грошей в суспільстві. Вдалою ідеєю є створення зручної та гнучкої системи управління такими фінансовими установами, тобто банками. Адже сьогодні не кожен може власноруч зареєструвати аккаунт в новому банку, відкрити в ньому власний рахунок, передавати гроші та контролювати їх кількість.

Банк — це вид кредитної фінансової установи, основною метою якої є контроль та управління грошима за допомогою різних пропонованих послуг, таких як зберігання великих сум грошей, проведення фінансових операцій або надання позик чи кредитів, серед інші. Звичайна практика банку — збирати капітал у різних фізичних осіб або компаній, які довіряють йому та ресурсам, заощаджуючи або перевіряючи рахунки. У той же час банк функціонує як просто ще одна компанія і багато разів має власні кошти. Звичайно, також із власною бізнес-ідеєю, коли справа стосується кредитних чи інших операцій у сфері фінансів.

					ІТ61.050БАК.004 ПЗ	Лист
						5
Змн.	Лист	№ докум.	Підпис	Дата		

У наш час функції та можливості банку розширюються з кожним днем. Все більше людей мають змогу виконувати різні операції та отримувати певні послуги відповідних установ. Але, незважаючи на це, не кожен має змогу особисто долучитися до цього процесу по різних причинах.

Сьогодення дає нам змогу задовольняти майже будь які потреби за допомогою різних додатків та Інтернету, проте, як виявилось, навіть у такий час багато банків залишаються без відповідного програмного забезпечення та змушують своїх клієнтів вичікувати черги задля того, щоб задовільнити якісь повсякденні потреби. Що там казати, я сам не раз відчував на собі потребу полегшення процесу реєстрації рахунків та операцій над ними за допомогою сучасних ресурсів.

Завданням проєкту є дослідження статистичних даних, пов'язаних з банківськими додатками України, специфіки реєстрації та управління банківськими рахунками, а також побудова програмного забезпечення, що спросить цей процес та стане основою для подальшого розвитку банківської системи в цілому. Перед початком роботи було проаналізовано відомі програми-аналоги в Україні та порівняв їх. Після їх порівняння було зроблено висновок про головні ознаки успішного продукту та популярні недоліки, які будуть уникнені.

Дана програма має зберігати інформацію про банки, їх клієнтів та працівників, банківські рахунки та транзакції. Кожен користувач повинен мати змогу зареєструватись та увійти під своїм логіном та паролем. Також має бути реалізовано перегляд даних про конкретного клієнта, зміну реєстраційних даних цього клієнта та додавання нових даних. Кожен клієнт повинен мати змогу переглядати інформацію про банки, в який його зареєстровано (номер телефону, адресу, назву) а також список працівників цього банку та, відповідно, інформацію про них (ім'я, телефон). Має бути реалізована можливість перегляду власних карт для клієнта. Список карт має сортуватись за різними показниками, наприклад за кількістю коштів на ній чи датою створення. Для кожної карти клієнт повинен мати змогу переглянути дані про цю карту (дату створення, термін дії, номер карти,

					IT61.050БАК.004 ПЗ	Лист
						6
Змн.	Лист	№ докум.	Підпис	Дата		

поточну кількість коштів тощо) а також здійснювати різні транзакції з нею (переказ коштів на іншу карту, поповнення рахунку, зняття коштів з рахунку тощо). Важливим є реалізація зберігання історії транзакцій для кожної карти кожного клієнта. Для реєстрації нового рахунку клієнт має зробити запит до менеджера про створення нової карти в деякому банку.

Актуальність теми: переважна відсутність спеціалізованих застосунків серед невеликих банків України.

Об'єкт дослідження: автоматизація реєстрації банківських рахунків та виконання фінансових операцій з ними.

Предмет дослідження: внутрішньобанківська система для реєстрації та управління рахунками.

Мета: дослідити сучасного становища банківських додатків в Україні та створити систему, що буде універсальною для різних банківських установ та дозволить користувачам власноруч поводити більшість фінансових операцій над рахунками.

Практичне значення: розроблюваний додаток надає клієнтам банку можливість власноруч проводити більшість фінансових операцій та реєстрації нових рахунків всередині банку.

Структура проєкту:

— у розділі 1 представлено актуальність проблеми, наведені деякі дослідження що підтверджують її, збір статистики для вибору основних вимог до системи, аналіз ринку та додатків-аналогів;

— у 2 розділі наведений макет розроблюваної системи, були визначені та описані основний функціонал системи, побудовано основні алгоритми, що будуть використовуватись у програмному забезпеченні та спроектовано базу даних для збереження інформації системи;

— у 3 розділі описано реалізацію системи, наведено опис основних класів системи, опис реалізації бази даних для збереження інформації системи та представлені деякі сценарії роботи програми;

					ІТ61.050БАК.004 ПЗ	Лист
						7
Змн.	Лист	№ докум.	Підпис	Дата		

- у 4 розділі описане тестування програмного забезпечення, наведено звіт до 3х видів тестування системи;
- у 5 розділі наведено інструкції для користувачів з ілюстрацією інтерфейсу додатку.

					ІТ61.050БАК.004 ПЗ	Лист
						8
Змн.	Лист	№ докум.	Підпис	Дата		

# 1 ОПИС ТА АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ

## 1.1 Загальні відомості

Банк проводить різноманітні види операцій з коштами та цінними паперами та надає фінансові послуги владі, юридичним та фізичним особам. По суті банк — це комерційна юридична особа, яка:

- створена з ціллю отримання вигоди;
- має право здійснювати різноманітні юридичні операції з коштами клієнтів-користувачів цих послуг;
- має особливе право на залучення грошових коштів фізичних та юридичних осіб з ціллю їх подальшого користування від свого імені;
- має право на відкриття та ведення банківських рахунків фізичних та юридичних осіб;
- діє на основі спеціального дозволу (ліцензії), що її надає відповідні державні органи;
- не має права проводити страхову, торгівельну та виробничу діяльність (таку як друк паперових грошей).

Банки є життєво важливою частиною економіки, адже вони надають дуже важливі фінансові послуги як бізнесу, так і звичайнісіньким споживачам. Як постачальник фінансових послуг, вони надають своїм клієнтам безпечне місце для зберігання готівки та електронних коштів. Завдяки різноманітним типам рахунків, таких як чекові та ощадні рахунки та депозитні сертифікати (компакт-диски), ви можете проводити звичайні банківські операції, такі як депозити, зняття коштів, запис чеків та оплата рахунків. Ви також можете заощадити свої гроші та заробити відсотки від своїх інвестицій.

Банки також надають кредитні можливості людям та корпораціям. Гроші, які ви депозитуєте в банку — короткострокові готівкові кошти — використовуються для позики іншим на довгострокову заборгованість, наприклад, автокредитування,

					ІТ61.050БАК.004 ПЗ	Лист
						9
Змн.	Лист	№ докум.	Підпис	Дата		

кредитні картки, іпотеку та інші боргові засоби. Цей процес допомагає створити ліквідність на ринку, що створює гроші та продовжує постачання.

Як і будь-який інший бізнес, метою банку є отримання прибутку для його власників. Для більшості банків власники є їх акціонерами. Банки роблять це, стягуючи більше позик за позики та інші борги, які вони видають позичальникам, ніж те, що вони платять людям, які користуються своїми накопичувальними засобами. Використовуючи простий приклад, банк, який сплачує 1% відсотків на ощадні рахунки та нараховує 6% відсотків за позики, отримує прибуток у 5% для своїх власників.

Банки бувають різні за розмірами залежно від місця їх знаходження та кого вони обслуговують — від невеликих установ, заснованих на громаді, до великих комерційних банків. Незважаючи на те, що традиційні банки пропонують як цегельне та будівельне розташування, так і Інтернет-присутності, на початку 2010-х років з'явилася нова тенденція в Інтернет-банках. Ці банки часто пропонують споживачам більш високі процентні ставки та менші комісії. Зручність, процентні ставки та збори — це деякі фактори, які допомагають споживачам визначитися з бажаними банками.

## 1.2 Дослідження та статистика банківських додатків в Україні

Створюючи будь який програмний продукт, необхідно передусім проаналізувати потреби суспільства, адже без цього створений додаток не буде мати ніякої користі. Тому в цьому випадку, маркетинговий аналіз дуже важливий, оскільки він дає можливість розкрити позиції на ринку, знати, що думають клієнти про таку продукцію, покращує вашу платформу для спілкування та обміну повідомленнями з клієнтами.

Насправді маркетинговий аналіз — це один із ключових методів підвищення якості створюваного додатку, рівня задоволеності майбутніх клієнтів та знання, які

					ІТ61.050БАК.004 ПЗ	Лист
Змн.	Лист	№ докум.	Підпис	Дата		10



фактори впливають на успішність розроблюваного продукту. Ось перелік найбільш важливих причин проведення маркетингового аналізу:

- розуміння ринку та клієнтів;
- визначення конкурентоспроможності такого програмного забезпечення;
- розуміння ціннісної пропозиції створюваного додатку;
- розуміння потреб споживачів у функціоналі системи, які можливості мають більший попит;
- виявлення засобів покращення якості продукту;
- виявлення засобів підвищення популярності продукту;
- знання, які можливості існують у межах ринку.

З мінливим ставленням до клієнтів та зростанням економічного різноманіття вирішальне значення має глибоке знання того, як працює ваш ринок електронного банкінгу. В ідеалі аналіз маркетингу — це систематичний спосіб збору статистики для розвитку бізнесу у банківській сфері. Це забезпечує впевненість, що маркетингові стратегії працюють відповідно до очікувань. Було досліджено основні елементи для проведення маркетингового аналізу.

Аналіз ринку та цільових клієнтів: Хоча маркетинговий аналіз має більш широку сферу застосування, його можна звужити до конкретних цілей. У цьому сенсі аналіз повинен бути зосередженим. Це означає, що потрібно провести аналіз на конкретному ринку та клієнтах, в даному випадку — на ринку цифрового банкінгу. Клієнтами ж цього ринку сьогодні є чи не кожен громадянин, тому в даному випадку корисним є дослідження навіть думки близького оточення.

Аналіз товарів і цін: Цей аспект стосується доцільності даного продукту на новому ринку або вдосконалення продукції на існуючому ринку. Так, потрібно використати дані, аби визначити, що майбутні клієнти даного додатку думають про його можливості, які вбачають шляхи вдосконалення та наскільки взагалі їм важливий розроблюваний функціонал. Проте питання ціни наразі не таке важливе.

Аналіз продажів: Для вже наявних аналогів аналіз продажів — це те, що допоможе розкрити цінні дані про ціни даного ринку, відкоригувавши їх або

					IT61.050БАК.004 ПЗ	Лист
Змн.	Лист	№ докум.	Підпис	Дата		11

запропонувавши знижки. Тому важливим є дослідження не тільки думки клієнтів про власний продукт, а ще й думку клієнтів інших додатків-конкурентів на сучасному ринку.

Канал розповсюдження: Однією з найсильніших маркетингових стратегій є планування способів пошуку майбутнього програмного забезпечення та підключення до клієнтів. У рамках аналізу необхідно визначити, які канали будуть використовуватися для розповсюдження даної продукції чи послуг вашим потенційним клієнтам.

Враховуючи специфіку об'єкта дослідження, вдалим рішенням є аналіз проведених досліджень з електронного банкінгу. Метою аналізу цих досліджень є підтвердження чи спростування потреби Українців у внутрішньобанківській системі локального управління рахунками.

Банки України все більше освоюють сучасні технології, намагаючись віднайти більш тісний зв'язок з своїми клієнтами, однак, як виявилось, переважна більшість українських банків все ще не користується додатками для полегшення цього невід'ємного процесу.

### 1.2.1 Дослідження Мінфіну

Наприкінці 2019 року «Мінфін» проаналізував кількість банків, що вже користуються додатками для полегшення взаємодії з клієнтами. Для збору даних статистика вони використовували найвідоміші платформи для встановлення додатків Google Play та AppStore, сайти відповідних банківських установ а також збирали відомості в контакт-центрах.

Це дозволило їм отримати інформацію щодо 77 українських банків. Як виявилось, програмним забезпеченням для клієнтів користуються лише 38 українських банків, тому створення універсальної платформи для операцій за рахунками є правильним рішенням в сучасних реаліях (рисунок 1.1). [13]

					IT61.050БАК.004 ПЗ	Лист
						12
Змн.	Лист	№ докум.	Підпис	Дата		



Рисунок 1.1 — Діаграма наявності додатку банків України

Як видно з рисунку 1.1, майже половина банків України не мають власного додатку для полегшення фінансових операцій.

### 1.2.2 Дослідження Mastercard

Дещо раніше, влітку 2019 було проведено дослідження Mastercard Digitalization Trust Survey 2019 [11]. Воно підтвердило, що все більше українців-власників банківських рахунків переходять на цифровий варіант банкінгу (рисунок 1.2).

Так, 87% українців надають перевагу цифровим сервісам для здійснення банківських платежів, 58% повсякденно користуються платіжними додатками (що є одним з найвищих показників серед опитаних країн Європи).



Рисунок 1.2 — Результати дослідження Mastercard [12]

### 1.2.3 Аналіз досліджень

Проаналізувавши дані досліджень маємо наступну ситуацію: українці надають перевагу електронним ресурсам, проте такими забезпечують своїх клієнтів далеко не всі банківські установи. Можна зробити висновок щодо доцільності розробки програмного забезпечення для управління банківськими рахунками.

Також статистика показує, що багато хто, маючи можливість користуватись додатками, не робить цього з різних причин. Ряд причин, за яких, можливо, не всі мають змогу користуватись бажаним програмним забезпеченням для роботи з коштами:

— відсутність необхідного обладнання. Далеко не кожен українець має комп'ютер, через який можна отримати доступ до веб сайтів відповідних установ. До того ж, навіть сьогодні далеко не в кожного є доступ до інтернету, необхідного для ефективного використання веб ресурсів;

— відсутність мотивації. Багато людей навіть у сучасному світі не розуміють переваг цифрового програмного забезпечення над звичними стаціонарними процесами. Економія витраченого часу та енергії занадто велика, проте не всі розуміють наскільки доступним стало програмне забезпечення для різних сфер звичної життєдіяльності;

— складність у користуванні. Більшість потенційних користувачів додатку так і не стають реальними клієнтами через страх та невмінні користуватись тим чи іншим програмним забезпеченням. Задля уникнення такої проблеми необхідним є створення детальної та зрозумілої інструкції для користувача. Це неодмінно збільшить кількість клієнтів продукту та підвищить доступність розроблюваного програмного забезпечення;

— відсутність інформування про існування такого програмного забезпечення. Як виявилось, навіть у сучасному світі не всі знають про існування того чи іншого забезпечення для спрощення звичних процесів повсякденного

життя. Саме тому необхідним є чітко сформулювати актуальність проблеми та мету створення саме такого програмного забезпечення.

### 1.3 Власне дослідження та збір статистики

Ринок банківських додатків в Україні вивчений не досконало, тому проблема дослідження потреб користувачів буде актуальною ще довго. Така інформація є дуже важливою при розробці будь-якого додатку, потенційними користувачами якого мають стати широкі маси. Враховуючи те, що мова йде про управління банківськими рахунками, клієнтом даного додатку може стати чи не кожен оточуючий. Тому проведення опитування оточення є доцільним та аналіз результатів має неабияке значення у розробці даного продукту при визначенні його функціоналу, структури та доцільності загалом.

Для збору інформації було обрано такі пункти, що допоможуть визначити основні напрями при розробці продукту та дадуть розуміння потреб сучасного користувача:

- зацікавленість в користуванні програмним забезпеченням для управління рахунками в банку;
- розуміння актуальності створення такого додатку;
- розуміння ринку;
- розуміння враженнями від додатків-аналогів;
- зрозуміти думку майбутніх клієнтів про переваги та недоліки успішних аналогів;
- доцільність реалізації того чи іншого функціоналу;
- важливість простого інтерфейсу;
- потребу у створенні універсальної платформи, що могла б підтримувати декілька банків одночасно;
- потребу у можливості відкрити декілька банківських рахунків в рамках одного банку.

					ІТ61.050БАК.004 ПЗ	Лист
						15
Змн.	Лист	№ докум.	Підпис	Дата		

Для проведення більш детального аналізу проблематики було прийнято рішення провести власне дослідження. За допомогою сервісу Google Forms була зібрана статистика щодо потреби користувачів в рамках досліджуваного об'єкту.

Як бачимо з рисунку 1.3, серед опитаних респондентів 65% відчують дискомфорт від перебування в чергах відділення банку, однак лише 28,3% з них користуються додатком.

Метою є зменшення кількості людей, що страждають від черг завдяки створенню локальних автоматизованих сервісів та збільшення кількості користувачів.

Також, згідно з результатами опитування (рисунок 1.4), 79,7% респондентів краще б власноруч скористались програмним забезпеченням, а не допомогою працівника банку у відділенні.

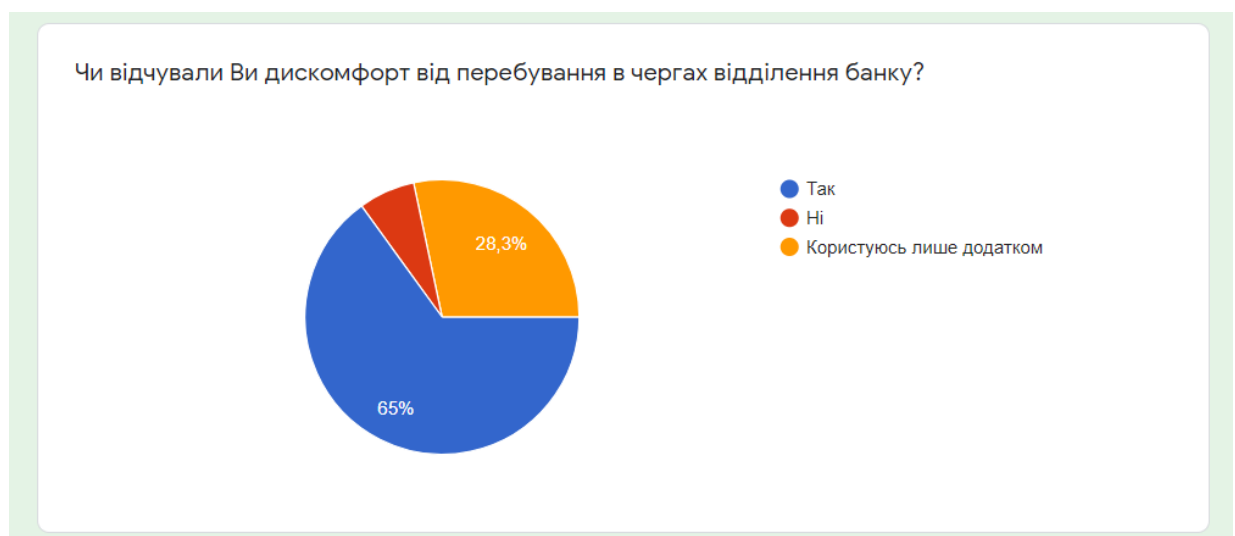


Рисунок 1.3 — Ілюстрація результату опитування на рахунок дискомфорту перебування в чергах

Це говорить про те, що більшість людей готові користуватись додатком, проте, як стало відомо раніше, далеко не усі банківські установи мають відповідне програмне забезпечення для задоволення відповідних потреб своїх клієнтів.

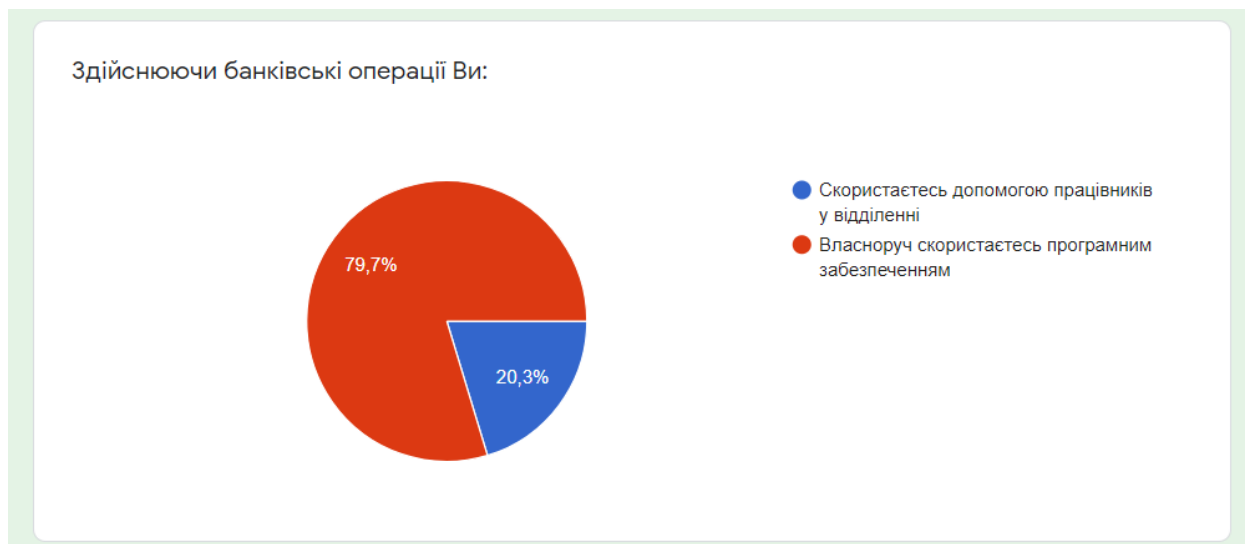


Рисунок 1.4 — Ілюстрація результатів опитування щодо вибору між працівником банку та відповідним програмним забезпеченням.

Також необхідно звернути увагу та проаналізувати той факт, що все ж 20,3% надають перевагу допомозі банківського працівника. Необхідним є створення простого та зрозумілого інтерфейсу користувача для даного програмного забезпечення. Так більша кількість користувачів матиме змогу взяти участь у технічному розвитку банківської сфери та долучиться до процесу цифрового банкінгу.

Важливим є також дослідження потреб користувачів у тому чи іншому функціоналі, тому в опитування були додані деякі запитання, що стосуються безпосередньо опцій самого програмного забезпечення.

Так було виявлено, що близько 60% потенційних користувачів надають перевагу можливості зберігати капітал в рахунках декількох банків одночасно (рис. 1.5).

Враховуючи ці результати, доцільним рішенням буде додати можливість реєструвати банківські рахунки в декількох банків на одного клієнта. Решта майже 40% потенційних користувачів надають перевагу одному банку та мають рахунки тільки в ньому, тому кількість рахунків не буде ніяк впливати на становище користувача та його можливості в системі.

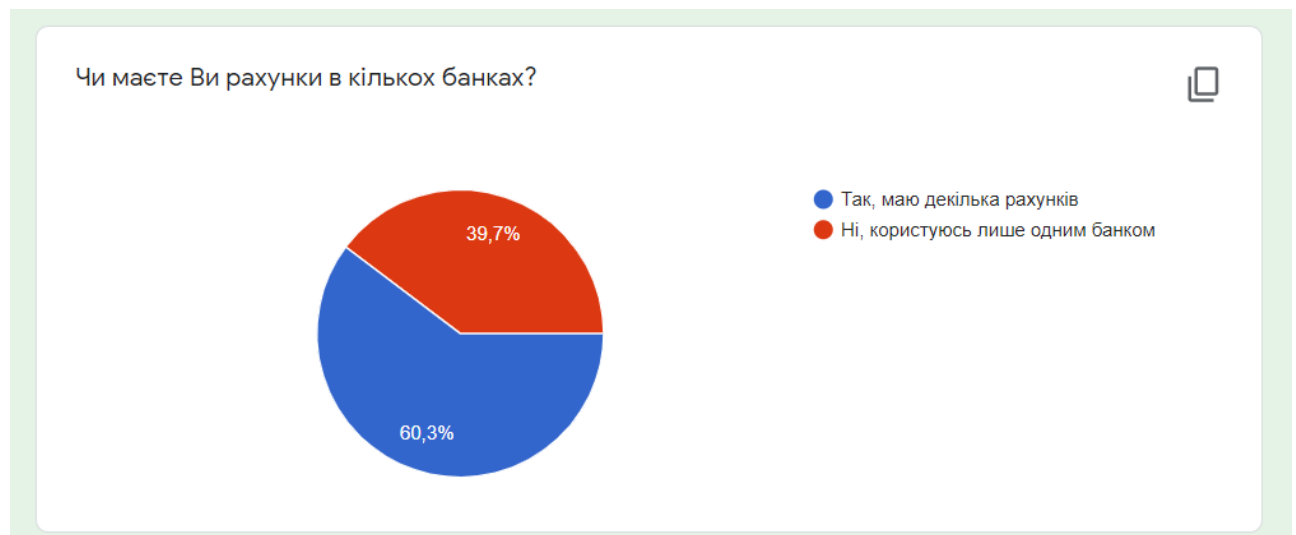


Рисунок 1.5 — Ілюстрація результатів опитування щодо наявності у користувачів рахунків у декількох банках.

Також опитування показало, що близько 99% користувачів користуються переказами коштів на карти інших банків (рис. 1.6). З них регулярними є перекази 25% клієнтів банків, решта користуються таким функціоналом лише іноді.



Рисунок 1.6 — Ілюстрація результатів опитування щодо частоти переказів на карти інших банків.

Враховуючи результати статистичних та сучасні тенденції, доцільним буде додати можливість переказу коштів на карти інших банків.



Проаналізувавши відповіді користувачів на дані запитання, можна зазначити декілька отриманих факторів:

— є потреба у пошуку альтернативи чергам у банківському відділенні (такою альтернативою і стане розроблюваний додаток);

— є потреба у програмному забезпеченні, що спростить роботу працівникам відділення;

— все ще є потреба у допомозі працівників відділення, що може бути зумовлена складністю аналогів у користуванні.

#### 1.4 Розвиток в умовах пандемії COVID-19

В наш час у світі склалася ситуація, в умовах якої суспільство вимушене залишатися вдома. Через це активізувалась розробка та розвиток додатків, що допомагають здійснювати повсякденні дії уникаючи безпосереднього контакту з іншими людьми. Банківську систему це не обійшло стороною. І хоч деякі банківські відділи продовжили свою роботу за обмеженим графіком, більшість людей не мають змоги повноцінно користуватись стаціонарними фінансовими установами.

В таких реаліях люди вимушені все більше вдаватись до використання цифрових ресурсів, тому стрімкого розвитку набули електронні додатки в усіх сферах життя. Банк не став виключенням, адже люди щодня проводять перекази, здійснюють покупки товарів та послуг. І, аби мінімізувати контакт з працівниками банківських установ, необхідним є створення відповідного програмного забезпечення, яке б давало змогу власноруч проводити необхідні банківські операції.

					IT61.050БАК.004 ПЗ	Лист
Змн.	Лист	№ докум.	Підпис	Дата		19

## 1.5 Огляд аналогів

Сьогодні кожного з нас оточує безліч додатків, цифрові сервіси використовують чи не у кожній сфері життя. Банківська система одна зі сфер, де використання відповідного програмного забезпечення є дуже доцільним, а іноді необхідним. Чи не щодня створюються та розвиваються нові й нові додатки інтернет банкінгу, тому аби конкурувати достойно, треба проаналізувати аналоги розроблюваної системи та зробити висновки про їх успішність, недоліки та переваги.

Саме тому є доцільність вивчити сучасний ринок та перейняти чужий досвід для більш успішного розвитку створюваного програмного забезпечення.

Аналізуючи програми-аналоги банківської сфери потрібно звернути увагу на такі критерії:

- зручність у використанні;
- збереження інформації;
- приємність візуального оформлення;
- простота інтерфейсу;
- низький рівень системних вимог;
- складність пошуку функції переказу коштів;
- складність створення нового рахунку;
- можливість переказу коштів на карту іншого банку;
- складність редагування зареєстрованих даних

Я вирішив проаналізувати додатки одних з найпопулярніших банків країни, а саме «Privat24», «Альфа-Банк».

### 1.5.1 Аналіз додатку «Privat24»

«Privat24» (рис. 1.7) — це додаток для керування своїм рахунком в банку «ПриватБанк». «ПриватБанк» є чи не найбільшим банком України. Він займає

					IT61.050БАК.004 ПЗ	Лист
Змн.	Лист	№ докум.	Підпис	Дата		20

перші позиції по всім фінансовим показникам в даній сфері. За деякими даними, саме він становить четверту частину усієї банківської системи країни та обслуговує третю частину всіх внесків населення країни.

ПриватБанк упродовж довгого часу користується попитом та довірою громадян та не тільки, тому додаток для користування цим банком неодмінно є взірцем для створення аналогу. До того ж 16 травня 2019 року другий рік поспіль Приват24 отримав нагороду FinAwards 2019 в номінації «Кращий інтернет-банк». Саме тому «Privat24» має стати одним з основних при аналізі успішних додатків в даній сфері.

Після аналізу веб версії додатку, можна виявити основні переваги даного продукту:

- адаптивність інтерфейсу;
- інформація про створені рахунки на першій сторінці;
- доступність можливості поповнення мобільного рахунку та переказу коштів на іншу карту;
- зрозумілі назви основних розділів меню;
- спокійні кольори візуального оформлення;
- історія транзакцій;
- інформація про ціну валют у видному місці;
- багато функціоналу, поділеного на розділи.

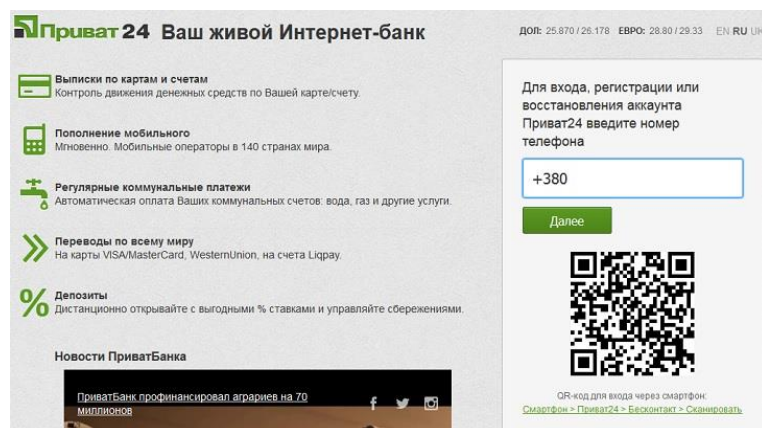


Рисунок 1.7 — WEB-додаток «Privat24» [14]

Також в ході вивченні додатку «Приват24», були виявлені деякі недоліки даного продукту:

- наявність дещо зайвої інформації;
- нагромаджений інтерфейс (рис. 1.8);
- перевантаження функціоналом;
- тусклі кольори в оформленні;
- деякі розділи незрозумілі, адже не мають підпису та мають лише неінформативне зображення;
- комбінування декількох прийомів UIUX дизайну (деякі пункти меню названі текстом, деякі — зображенням (рис. 1.9), забагато різних шрифтів в одному місці);
- забагато посилань на одній сторінці;
- наявність реклами.

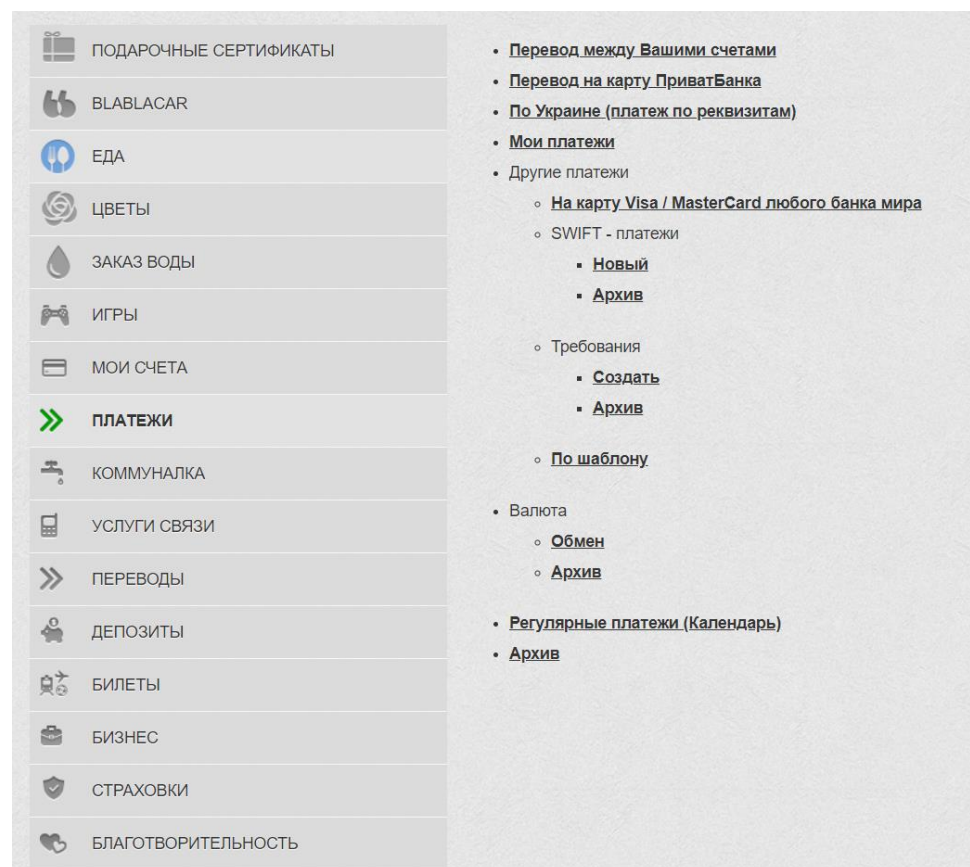


Рисунок 1.8 — Меню услуг «Privat24» [14]



Рисунок 1.9 — Розділи веб версії «Privat24» [14]

### 1.5.2 Аналіз додатку «Альфа-Банк»

Альфа-Банк — український комерційний банк з міжнародним капіталом. Банк входить до топ-10 країни по розміру активів та займає лідируючі позиції у всіх сегментах ринку країни.

«Альфа-Банк» — це додаток керування рахунками банку «Альфа-Банк», що дає змогу виконувати повсякденні фінансові операції.

Після ознайомлення з веб версією додатку «Альфа-Банк» було виявлено основні переваги даного продукту:

- приємне та сучасне візуальне оформлення;
- адаптивність інтерфейсу;
- основна функція переказу коштів доступна відразу після авторизації на головній сторінці додатку (рис. 1.10);
- зручна навігація по розділам;
- лаконічний та інтуїтивно зрозумілий інтерфейс з акцентами на важливих функціях;
- помітні контактні дані для допомоги;
- вибір основного кольору застосунку.

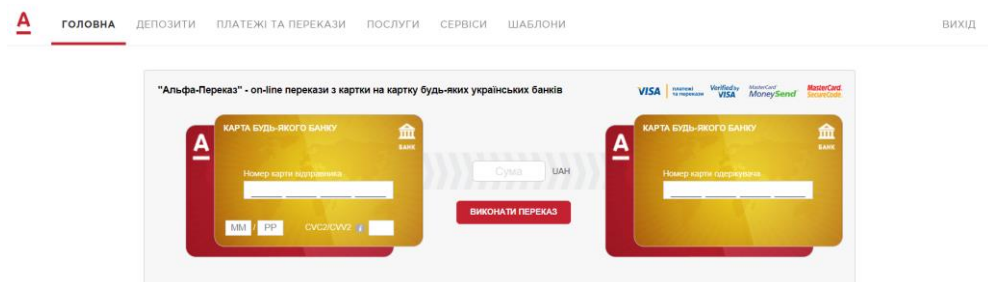


Рисунок 1.10 — Головна сторінка веб версії додатку «Альфа-Банк» [15]

					IT61.050БАК.004 ПЗ	Лист
Змн.	Лист	№ докум.	Підпис	Дата		23

При користування веб застосунком «Альфа-Банк» також були виявлені деякі недоліки даного продукту:

- складність пошуку історії транзакцій;
- замалий шрифт для ілюстрації важливої інформації;
- не достатньо зрозумілі назви розділів застосунку;
- перевантаження функціоналом в деяких розділах додатку;
- задовге очікування при переході між розділами.

### 1.5.3 Порівняння

Переглянувши додатки-аналоги, можна проаналізувати критерії успішності досліджених продуктів та важливість уникнення певних недоліків.

Так, виявленою перевагою «Privat24» швидкий доступ до головної інформації та деяких функцій, проте нагромадження інтерфейсу робить користування цим додатком незручним для недосвідченого користувача.

«Альфа-Банк» здивував приємним візуальним оформленням та доступністю основних функцій, проте складність пошуку деяких можливостей та іноді задовгі очікування запиту роблять цей додаток не надто зручним для користування пересічним громадянином.

Для більш інформативного аналізу, на основі зібраної інформації була створена порівняльна таблиця основних критеріїв банкододатків (таблиця 1.1).

Таблиця 1.1 — Порівняльна таблиця банківських додатків «Privat24» та «Альфа-Банк»

Критерій	Веб версія Privat24	Веб версія Альфа-Банк
Зручність у використанні	-	+

Критерій	Веб версія Privat24	Веб версія Альфа-Банк
Збереження інформації	+	+
Приємне візуальне оформлення	-	+
Простота інтерфейсу	-	-
Низький рівень системних вимог	+	-
Легкість пошуку функції переказу коштів	+	+
Легкість створення нового рахунку власноруч	-	-
Можливість переказу коштів на карту іншого банку	+	+
Простота редагування зареєстрованих даних	+	-

### Висновки до розділу

Проаналізувавши всі дані та статистику, можна зробити висновок про актуальність проблеми та потребу створення банківського додатку для управління рахунками. Також на основі проведених досліджень можна виокремити основні критерії успішного продукту в даній сфері:

- простота та зручність у використанні;
- можливість зберігати інформацію;
- доступність пошуку функції переказу коштів;
- можливість переказу коштів на картки інших банків;
- можливість редагування зареєстрованих даних.

## 2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ СИСТЕМИ

### 2.1 Вибір функціоналу системи

Дослідження, описане в пункті 1.3 показало основні напрямки розвитку розроблюваного додатку, популярні та необхідні функції такої системи. Аналіз результатів цього дослідження дало змогу побудувати діаграму прецедентів основного функціоналу додатку.

Діаграма прецедентів представлена у кресленику IT61.050БАК.004 Д1, більш детальний опис основних функцій наведений в таблицях 2.1 — 2.3.

Таблиця 2.1 — Варіант використання UC001

Назва	Реєстрація в новому банку
Опис	Користувач подає запит на реєстрацію в новому обраному банку
Учасники	Користувач у ролі клієнта
Передумови	Користувач пройшов автентифікацію до системи. Інформація про банк записана у базі даних, представлена у системі та доступна користувачеві
Результат	Користувач записаний до переліку користувачів банку в базі даних, у вікні даних про клієнта з'являється інформація про новий банк.



Основний сценарій	Користувач надає запит на реєстрацію в обраному банку. Система оброблює запит та додає відповідні дані до бази даних.
-------------------	--

Таблиця 2.2 — Варіант використання UC005

Назва	Запит на реєстрацію нового рахунку
Опис	Користувач подає запит на реєстрацію нового рахунку в конкретному банку
Учасники	Користувачі у ролі клієнта та менеджера
Передумови	Користувачі пройшли автентифікацію до системи, клієнт та менеджер зареєстровані в даному банку. Відповідна інформація записана до бази даних.
Результат	На ім'я клієнта зареєстрований новий рахунок в базі даних, у вікні даних про клієнта з'являється інформація про цей рахунок.
Основний сценарій	Клієнт надає запит на реєстрацію рахунку в обраному банку. Система оброблює запит та передає його на менеджера. Менеджер підтверджує реєстрацію даного рахунку на ім'я даного клієнта. Користувач отримує повідомлення про успішну реєстрацію рахунку в новому банку.

Таблиця 2.3 — Варіант використання UC007

Назва	Здійснення переказу кошт
Опис	Користувач подає запит на здійснення переказу кошт на вказаний рахунок
Учасники	Користувач у ролі клієнта
Передумови	Користувач пройшов автентифікацію до системи. Він зареєстрований в банку, на нього зареєстровано рахунок. Відповідна інформація записана до бази даних.
Результат	На обраному рахунку клієнта кількість кошт зменшилась на вказане значення, на вказаному для переказу рахунку кількість кошт збільшилась на це значення. Відповідні дані в базі даних оновлено.
Основний сценарій	Клієнт обирає рахунок, з якого хоче здійснити переказ, вказує рахунок на який необхідно переказати кошти, відповідні дані своєї карти для підтвердження прав власника та кількість кошт, що візьмуть участь в операції. Після успішного переказу користувач отримує повідомлення про здійснення операції.

Така реалізація функціоналу буде зрозумілою для користувачів та логічною для розробки програмного забезпечення.

## 2.2 Розробка деяких алгоритмів

Для швидкодії та оптимізації програми необхідним є побудова алгоритмів для різних аспектів системи. На рисунку 2.1 представлено послідовність дій системи та користувача в момент реєстрації.

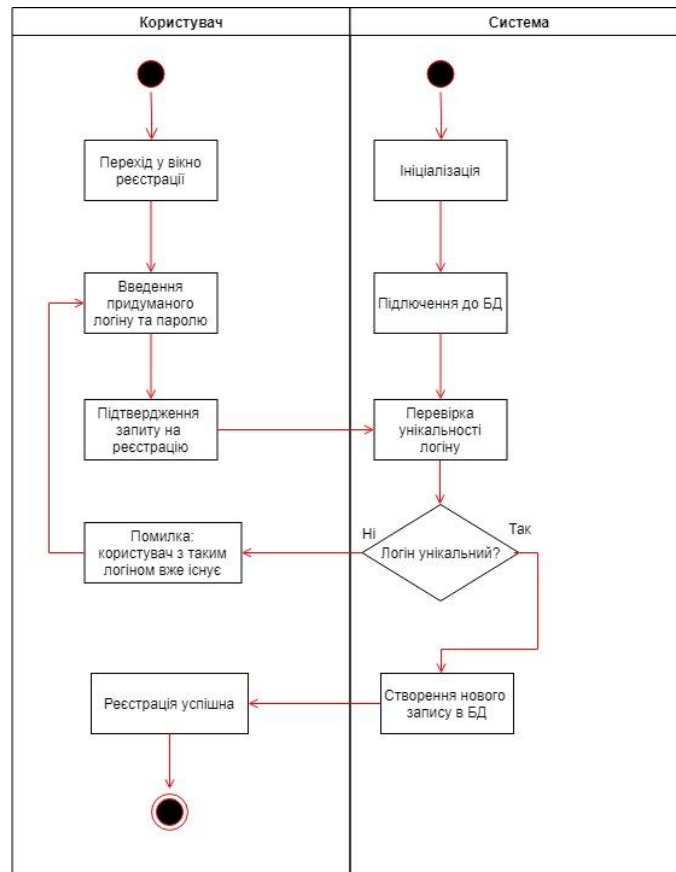


Рисунок 2.1 — Діаграма активності процесу реєстрації

Вибір такого алгоритму дій є необхідним, адже така послідовність дій задовольняє усі вимоги до даного процесу, а саме:

- перевірка унікальності користувача в системі;
- запис усіх даних до БД;
- відповідне інформування користувача.

Швидкодія є чи не найважливішим показником для операції переказу коштів, тому розробка оптимального алгоритму для цього процесу є важливим етапом в розробці проєкту.

### 2.3 Проєктування бази даних для системи

База даних — один з найважливіших аспектів у створенні системи для управління рахунками в банку. Адже надійне зберігання даних є чи не найважливішим критерієм успішного продукту даного роду.

Проєктування бази даних є важливим питанням, адже від цього залежить зручність редагування та гнучкість системи зберігання інформації.

В базі даних інформація моделюється у вигляді таблиць, що мають рядки та стовпці. Рядок дає інформацію про конкретний запис (об'єкт), а стовпець дає інформацію про одне поле (характеристику) об'єктів даної таблиці. Це допомагає зробити обробку та запити даних більш ефективними та зручними. У такому вигляді дуже легко отримати доступ, керувати, змінювати, оновлювати, контролювати та впорядковувати дані.

Для даного проєкту необхідним є проєктування бази даних, що буде містити наступні таблиці:

- користувачі системи;
- клієнти банку;
- банківські рахунки;
- менеджери банків;
- банки;
- історія транзакцій.

Для відображення користувачів системи було обрано структуру, зображену на таблиці 2.4.

Таблиця 2.4 — Логічна структура таблиці «Клієнти банку»

Назва поля	Тип	Опис
Логін	Текст	Логін користувача, що має бути унікальним та використовується при авторизації до системи та визначає роль в ній.
Пароль	Текст	Пароль користувача, що має бути секретним та використовується при авторизації до системи.
Ім'я клієнта	Текст	Ім'я користувача, що надає змогу ідентифікувати особу для проведення фінансових операцій, а також створення нових рахунків.
Прізвище клієнта	Текст	Прізвище користувача, що надає змогу ідентифікувати особу для проведення фінансових операцій, а також створення нових рахунків.
Банки	Список	Список банків, в яких зареєстровано даного клієнта
Рахунки	Список	Список рахунків, які зареєстровані на даного клієнта

Для відображення рахунків було обрано структуру, зображену на таб. 2.5

Таблиця 2.5 — Логічна структура таблиці «Банківські рахунки»

Назва поля	Тип	Опис
Термін дії карти	Дата	Дата та час закінчення терміну використання даного рахунку (карти). Зазвичай 5 років з моменту реєстрації.
Кошти	Число	Поточна кількість коштів певної валюти, присутня на даному рахунку.

Назва поля	Тип	Опис
Номер карти	Число	Номер карти, що використовується як ідентифікатор.
ПІН-код	Число	Секретний код карти, що використовується для підтвердження прав на користування даною картою для здійснення фінансових операцій з нею.
Банк	Об'єкт	Банк, в якому зареєстрований даний рахунок.
Клієнт	Об'єкт	Клієнь банку, на якого зареєстровано даний банківський рахунок.

Для відображення менеджерів було обрано структуру зберігання інформації, зображену на таблиці 2.6.

Таблиця 2.6 — Логічна структура таблиці «Менеджери»

Назва поля	Тип	Опис
Логін	Текст	Логін менеджера, що має бути унікальним та використовується при авторизації до системи та визначає роль менеджера.
Пароль	Текст	Пароль менеджера, що має бути секретним та використовується при авторизації до системи.
Ім'я	Текст	Ім'я менеджера, що надає змогу ідентифікувати особу.
Прізвище	Текст	Прізвище користувача, що надає змогу ідентифікувати особу.
Електронна адреса	Текст	Електронна поштова адреса менеджера для більш зручної комунікації та додаткових питань, що виникають у клієнтів.

Назва поля	Тип	Опис
Банк	Об'єкт	Банк, в якому зареєстровано даного менеджера як працівника.

Для відображення банків системи було обрано структуру зберігання даних, зображену на таблиці 2.7.

Таблиця 2.7 — Логічна структура таблиці «Банки»

Назва поля	Тип	Опис
Назва	Текст	Назва банку, використовується для ідентифікації банку клієнтами.
Контактний телефон	Число	Контактний номер телефону банку для детальної інформації та запитань.
Адреса	Текст	Фізична адреса відділення банку, для контактної інформації.
Клієнти	Список	Користувачі, які зареєстровані клієнтами в даному банку
Менеджери	Список	Менеджери, які зареєстровані працівниками даного банку

Для відображення історії транзакцій «Історія транзакцій» було обрано структуру зберігання даних, зображену на таблиці 2.8.

Таблиця 2.8 — Логічна структура таблиці «Історія транзакцій»

Назва поля	Тип	Опис
Кошти	Число	Грошове значення, що було переказано в рамках даної транзакції.
Дата та час	Дата та час	Дата та час підтвердження запиту на фінансову транзакції
Карта переказувача	Об'єкт	Карта клієнта, який ініціював запит на транзакцію.
Статус	Стан	Статус транзакції: ,скасована або недопущена за правилами банку, в очікуванні, успішно завершена.
Карта отримувача	Об'єкт	Карти отримувача переказаних коштів.

Така структура таблиць дозволить зручно записувати, змінювати та видаляти об'єкти з бази даних.

## 2.4 Вибір інструментів

Серверну частину реалізовано на платформі .NET мови C# для використання її об'єктно-орієнтованого синтаксису, ця мова використовується в багатьох великих організаціях, стартапах, а також початківцями. Вона зручна і має безкоштовне середовище розробки, що для мене, як для студента, є безумовною перевагою.

### 2.4.1 Вибір IDE

Інтегроване середовище розробки (англ. Integrated development environment або IDE) — програмне забезпечення, що допомагає розробникам створювати власне програмне забезпечення. Зазвичай IDE включають редактор для мови



програмування, засоби для налагодження програми, компілятори, інтерпретатори. В даному випадку використання IDE є необхідним і тому, що сучасні середовища, що підтримують орієнтацію на об'єкти, також включають зручні переглядачі для об'єктів та діаграми структури об'єктів.

Загалом, IDE має ще деякі функції, які допоможуть при розробці даного продукту, а саме:

- редактор: використовується для написання коду та кольорового виділення синтаксису для зручності;

- кодовий запит: необхідно ввести лише частину коду, після чого середовище підкаже варіанти написання подальшого коду, це прискорює процес створення додатків;

- налагоджувач: середовище розробки дає змогу спостерігати за кожним кроком програми під час її виконання для більш зручного пошуку логічних помилок;

- зручний інтерфейс: елементи меню та панелі допоможуть більш швидко впроваджувати зміни та гнучко налаштовувати проєкт, до того ж, це неабияк прискорює роботу.

Найпопулярнішою IDE для розробки програмного забезпечення з використанням мови програмування C# є Microsoft Visual Studio, тому при розробці даного проєкту доцільним є використання саме цієї IDE.

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів.

Було виокремлено деякі причини вибору саме Microsoft Visual Studio для створення даного проєкту:

- внутрішнє управління бібліотеками дозволяє керувати бібліотеками та розширеннями не виходячи з середовища;

— можливість синхронізації за допомогою ідентифікатора облікового запису Microsoft або GitHub дозволяє легко перекидати налаштування проєкту між кількома комп'ютерами;

— гнучкість цієї IDE дає змогу створювати будь-якого типу додатків, включаючи інформаційні системи;

— ілюстрування: інструменти для побудови графіків та діаграм для зручного ілюстрування та розробки додатків;

— розумний код доповнює код та здійснює рефакторинг за допомогою надсучасними AI.

Цей продукт також дозволяє розробляти програми з графічним інтерфейсом, в тому числі з підтримкою технології WPF. Саме через підтримку даної технології було обрано це середовище розробки. Для створення даного продукту було обрано останню версію даної IDE — Visual Studio 2017.

#### 2.4.2 Основні рішення щодо збереження інформації

База даних є необхідним компонентом для створення даної системи. Вона буде слугувати складом, який організовує, зберігає та управляє даними відповідно до обраної структури даних. Це також буде сукупністю великих обсягів даних, що зберігаються на пристрої тривалий час, організовуються, обмінюються та уніфікуються.

У найпоширеніших типах БД, як правило, дані моделюються у серію таблиць у вигляді рядків та стовпців, де рядок дає інформацію про конкретний запис (об'єкт), а стовпець дає інформацію про одне поле (характеристику) об'єктів даної таблиці. Це допомагає зробити обробку та запити даних більш ефективними та зручними. У такому вигляді дуже легко отримати доступ, керувати, змінювати, оновлювати, контролювати та впорядковувати дані.

База даних зазвичай вимагає комплексну програму для керування бази даних, або систему управління базою даних (СУБД). СУБД в даній системі буде

					ІТ61.050БАК.004 ПЗ	Лист
Змн.	Лист	№ докум.	Підпис	Дата		36

виконувати роль провідника між базою даних та програмою, дозволяючи їй отримувати, оновлювати та керувати записаною в БД інформацією. Крім того, СУБД допомагає контролювати БД шляхом забезпечення різними операціями управління, а саме:

- моніторинг продуктивності;
- гнучке налаштування структури інформації;
- резервне копіювання;
- відновлення даних.

Головним завданням БД в даній системі є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Для вирішення завдання збереження інформації для мого проєкту було обрано базу даних MySQL та СКБД MySQL. MySQL надає багатий набір функціональних можливостей, які підтримують безпечне середовище для зберігання, обслуговування і отримання даних. Вона характеризується великою швидкістю, стійкістю і простотою використання, була розроблена для підвищення швидкодії обробки великих баз даних.

Отже було сформовано перелік основних переваг обраної мною СКБД:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

### 2.4.3 Вибір інструменту для створення графічного інтерфейсу

В даному випадку найважливішим є простота та доступність інтерфейсу даної системи, адже метою створення даного програмного продукту є спрощення процесу управління банківськими рахунками. Основою такої програми мають бути прості, великі за розміром та помітні елементи інтерфейсу, щоб дати можливість будь-яким користувачам легко взаємодіяти зі створюваною системою. Тому, розробляючи інтерфейс даного додатку, буде звернено увагу щоб він був інтуїтивно зрозумілий будь-кому.

Для цього завдання було прийняте рішення обрати платформу WPF, що дозволяє створювати інтерфейси для програмних рішень та дає змогу гнучко його налаштовувати за допомогою XAML файлів.

Технологія WPF (Windows Presentation Foundation) це платформа для створення клієнтських додатків для настільних ПК, що підтримує широкий спектр функцій розробки додатків, включаючи моделі прикладних програм, ресурси, елементи керування, графіку, макет, інтерфейс тощо. Цей фреймворк є частиною .NET framework, тому WPF є доцільним вибором для використання разом з мовою програмування C#.

Як вже було зазначено, однією з важливих особливостей є використання мови декларативною розмітки інтерфейсу XAML, заснованого на XML, це дозволяє створювати насичений графічний інтерфейс, використовуючи або декларативне оголошення інтерфейсу, або код на мовах C#. Це стало основною причиною обрання мною саме цієї технології.

Отже основні переваги WPF:

- використання традиційних мов .NET платформи;
- можливість декларативного визначення графічного інтерфейсу за допомогою спеціальної мови розмітки XAML, що підвищує гнучкість при розробці інтерфейсу програмного забезпечення;

— незалежність від розміру екрану: оскільки в WPF всі елементи вимірюються в незалежних від пристрою одиницях, додатки на WPF легко масштабуються під різні екрани з різним розміром;

— нові можливості, яких складно було досягти в аналогічних WinForms, наприклад, створення тривимірних моделей, прив'язка даних, використання таких елементів, як стилі, шаблони, теми і ін., створення анімацій, прив'язка даних, стилі, шаблони, теми і багато іншого;

— створення додатків під безліч ОС сімейства Windows — від Windows XP до Windows 10.

### Висновки до розділу

Для розробки даного програмного забезпечення було обрано доцільний та зрозумілий функціонал. Було спроектовано основні алгоритми, що будуть використовуватись у розробці.

Також було обрано та проаналізовано інструменти для розробки даного проекту, а саме:

- мову програмування C#;
- інтегроване середовище розробки MS Visual Studio;
- СКБД MsSQL;
- платформу для розробки інтерфейсу WPF.

### 3 ОПИС ПРОГРАМИ

#### 3.1 Реалізація системи ролей

У системі запропоновано три ролі користувачів: client, worker та admin. Кожна з них має свої повноваження у системі.

При запуску додатку користувач відразу має пройти авторизацію, яка визначає роль користувача інформаційної системи для банків (клієнт банку, менеджер чи адміністратор). Для цього необхідно ввести логін та пароль у відповідні поля вікна.

Якщо користувач ще не зареєстрований у системі він має змогу натиснути кнопку «Зареєструватись». Після цього відкривається вікно реєстрації нового клієнта банку, де користувач має змогу ввести ім'я, прізвище, номер телефону, логін та пароль для авторизації. Після натискання кнопки «Зареєструватись» виконується вхід в систему під щойно зареєстрованим користувачем з роллю клієнта банку.

Менеджера банку має змогу зареєструвати тільки адміністратор системи. Він надсилає реєстраційні дані менеджеру, після чого той може увійти до системи з своєю роллю.

Щоб зайти до системи в ролі адміністратора потрібно ввести слова «admin» в поля логіну та паролю. Кодове значення для автентифікації адміністратора було обрано як експериментальне та може змінитись.

Після вводу реєстраційних даних необхідно натиснути кнопку «Вхід» після чого виконується вхід в систему за одним з трьох сценаріїв.

#### 3.2 Основні класи системи

Діаграма класів системи представлена в кресленику IT61.050БАК.004 ДЗ. В таблиці 3.1 наведений опис основних класів системи.

					IT61.050БАК.004 ПЗ	Лист
						40
Змн.	Лист	№ докум.	Підпис	Дата		

Таблиця 3.1 — Основні класи системи

Назва класу в системі	Опис
Base	Базовий клас для наслідування, містить в собі генерацію ідентифікатора, що буде використовуватись в дочірніх класах.
Bank	<p>Клас для реалізації об'єкту банку в системі.</p> <p>Містить список всіх банків системи, має поля адреси, назви та контактного телефону.</p> <p>В класі реалізуються методи додавання нового запису про банк до БД, редагування записаних даних та зчитування даних про запис у БД в таблиці банків.</p>
Card	<p>Клас для реалізації об'єкту рахунку (банківської карти) в системі.</p> <p>Містить список всіх карток в системі.</p> <p>Має поля номеру карти, ПІН-коду, кількості коштів та терміну дії карти.</p> <p>В класі реалізуються методи додавання нового запису про карту до БД, редагування записаних даних, зачитування даних з відповідної таблиці, генерації номеру карти та секретного ПІН-коду. Також клас реалізує базові фінансові операції, такі як переказ коштів на іншу карту.</p>
User	<p>Клас для реалізації універсального користувача системи, від якого будуть пронаслідувані конкретні ролі користувачів (клас client та клас worker).</p> <p>Зберігає інформацію про користувача, під іменем якого авторизовано даний сеанс. Має поля логіну та паролю користувача, а також імені та фамілії для ідентифікації.</p> <p>В класі реалізуються методи маніпуляцій з реєстраційними даними користувача.</p>

Назва класу в системі	Опис
Client	<p>Клас для реалізації об'єкту користувача системи у ролі клієнта банку.</p> <p>Містить список всіх клієнтів в системі.</p> <p>Клас є дочірнім від класу User, тому має аналогічні поля інформації про користувача</p> <p>В класі реалізуються методи додавання нового клієнту до БД, редагування записаних даних про клієнта та зачитування даних з відповідної таблиці.</p>
Worker	<p>Клас для реалізації об'єкту користувача системи у ролі менеджера банку.</p> <p>Містить список всіх менеджерів в системі.</p> <p>Клас є дочірнім від класу User, тому має аналогічні поля інформації про користувача та додаткове поле для цієї ролі, що є електронним адресом для контактної інформації.</p> <p>В класі реалізуються методи додавання нового менеджера до БД, редагування записаних даних про менеджера та зачитування даних з відповідної таблиці.</p>

### 3.3 Реалізація бази даних

Було розроблено реляційну базу даних MsSQL. Структурну схему реалізації бази даних наведено у кресленику IT61.050БАК.004 Д4.

В таблиці 3.2 наведений детальний опис типів та назви полів, що представлені у відношенні clients бази даних системи.



Таблиця 3.2 — Реалізація таблиці clients

Назва поля	Тип	Опис
ID	Guid	Унікальний автоматично згенерований ідентифікатор користувача
Login	VARCHAR(10)	Логін користувача, що має бути унікальним та використовується при авторизації до системи та визначає роль в ній.
Password	VARCHAR(10)	Пароль користувача, що має бути секретним та використовується при авторизації до системи.
Name	VARCHAR(15)	Ім'я користувача, що надає змогу ідентифікувати особу для проведення фінансових операцій, а також створення нових рахунків.
Surname	VARCHAR(15)	Прізвище користувача, що надає змогу ідентифікувати особу для проведення фінансових операцій, а також створення нових рахунків.

В таблиці 3.3 наведений детальний опис типів та назви полів, що представлені у відношенні cards бази даних системи.

Таблиця 3.3 — Реалізація таблиці cards

Назва поля	Тип	Опис
ID	Guid	Унікальний автоматично згенерований ідентифікатор рахунку.
Term	Datetime()	Дата та час закінчення терміну використання даного рахунку (карти). Зазвичай 5 років з моменту реєстрації.

Назва поля	Тип	Опис
Money	double	Поточна кількість коштів певної валюти, присутня на даному рахунку.
Number	INT	Номер карти, що використовується як ідентифікатор.
Pin	INT	Секретний код карти, що використовується для підтвердження прав на користування даною картою для здійснення фінансових операцій з нею.
Bank_id	Guid	Ідентифікатор банку, що має бути унікальним, та є посиланням на банк, в якому зареєстрований даний рахунок.
Client_id	Guid	Ідентифікатор клієнта банку, що має бути унікальним, та є посиланням на клієнта банку, на якого було зареєстровано даний банківський рахунок.

В таблиці 3.4 наведений детальний опис типів та назви полів, що представлені у відношенні workers бази даних системи.

Таблиця 3.4 — Реалізація таблиці workers

Назва поля	Тип	Опис
ID	Guid	Унікальний автоматично згенерований ідентифікатор менеджера
Login	VARCHAR(10)	Логін менеджера, що має бути унікальним та використовується при авторизації до системи та визначає роль менеджера.

Назва поля	Тип	Опис
Password	VARCHAR(10)	Пароль менеджера, що має бути секретним та використовується при авторизації до системи.
Name	VARCHAR(15)	Ім'я менеджера, що надає змогу ідентифікувати особу.
Surname	VARCHAR(15)	Прізвище користувача, що надає змогу ідентифікувати особу.
Email	VARCHAR(20)	Електронна поштова адреса менеджера для більш зручної комунікації та додаткових питань, що виникають у клієнтів.
Bank_id	Guid	Ідентифікатор банку, що має бути унікальним, та є посиланням на банк, в якому зареєстрований даний менеджер.

В таблиці 3.5 наведений детальний опис типів та назви полів, що представлені у відношенні banks бази даних системи.

Таблиця 3.5 — Реалізація таблиці banks

Назва поля	Тип	Опис
ID	Guid	Унікальний автоматично згенерований ідентифікатор банку
Name	VARCHAR(30)	Назва банку, використовується для ідентифікації банку клієнтами.
Phone number	INT	Контактний номер телефону банку для детальної інформації та запитань.
Adress	VARCHAR(50)	Фізична адреса відділення банку, для контактної інформації.

Для більш зручної реалізації зв'язку «багато до багатьох» між банками та клієнтами, було прийняте рішення створити розподільну таблицю з посиланнями на об'єкти відповідних таблиць. Реалізацію такого відношення в базі даних наведено в таблиці 3.6.

Таблиця 3.6 — Реалізація таблиці bank\_client

Назва поля	Тип	Опис
ID	Guid	Унікальний автоматично згенерований ідентифікатор клієнта банку
Bank_id	Guid	Ідентифікатор банку, що має бути унікальним, та є посиланням на банк, в якому зареєстрований даний клієнт.
Client_id	Guid	Ідентифікатор користувача системи, що має бути унікальним, та є посиланням на користувача, який зареєстрований як клієнт банку.

В таблиці 3.7 наведений детальний опис типів та назви полів, що представлені у відношенні history бази даних системи.

Таблиця 3.7 — Реалізація таблиці history

Назва поля	Тип	Опис
ID	Guid	Унікальний автоматично згенерований ідентифікатор транзакції
Money	DOUBLE	Кількість коштів, що були переказані в рамках даної транзакції.
Datetime	DateTime	Дата та час здійснення запиту на фінансову транзакцію

Назва поля	Тип	Опис
Card_id	Guid	Ідентифікатор карти клієнта, який ініціював запит на транзакцію.
Status	INT	Статус транзакції: — 0, якщо скасована або недопущена за правилами банку; — 1, якщо в очікуванні; — 2, якщо успішно завершена.
End_card_id	Guid	Ідентифікатор карти отримувача переказаних коштів.

Така структура таблиць дає змогу зручно записувати, змінювати та видаляти об'єкти з бази даних.

### 3.4 Структура вікон додатку

Схема структури вікон додатку відображено в кресленику IT61.050БАК.004 Д2. В таблиці 3.8 приведено більш детальний опис вікон та переходу між ними.

Таблиця 3.8 — Опис класів вікон додатку

Назва	Опис
MainWindow	Початкове вікно додатку. Тут користувач має змогу авторизуватись до системи або перейти на вікно реєстрації wnd_ClientRegistration.  При авторизації в ролі клієнта банку користувач переходить на вікно списку карт wnd_ClientMain.  При авторизації в ролі менеджера банку користувач переходить на вікно інформації про банк wnd_Worker_Main.

Назва	Опис
	При авторизації в ролі адміністратора системи користувач переходить на вікно списку банків системи wnd_AdminMain.
wnd_ClientMain	Вікно профілю користувача, який авторизувався у ролі клієнта банку. Вікно містить список зареєстрованих на нього рахунків. З цього вікна можна перейти до вікна здійснення транзакцій wnd_Transaction, вікна реєстрації нового рахунку в банку wnd_ClientNewCard або ж до вікна зміни реєстраційних даних wnd_ClientEdit. Також з цього вікна можна повернутись до вікна авторизації
wnd_ClientEdit	Вікно зміни реєстраційних даних для клієнта. З цього вікна можна повернутись до профілю клієнта wnd_ClientMain.
wnd_ClientRegistration	Вікно реєстрації користувача в ролі клієнта банку. Після успішної реєстрації повертає на вікно авторизації користувачів MainWindow.
wnd_ClientNewCard	Вікно реєстрації нового банківського рахунку для авторизованого клієнта. Після успішного запиту на реєстрацію нового рахунку користувач повертається на вікно профілю wnd_ClientMain.
wnd_Transaction	Вікно відображення історії транзакцій для обраної карти. Також тут реалізовано процес здійснення банківських операцій з обраною картою. З цього вікна можна повернутись до профілю клієнта wnd_ClientMain.
wnd_AdminMain	Вікно зі списком усіх банків та інформацією про кожен з них. Головне вікно адміністратора системи. З цього вікна можна перейти до реєстрації нового працівника банку

Назва	Опис
	wnd_AdminAddWorker або реєстрацію нового банку в системі wnd_AdminAddBank. Також з нього можна повернутись на вікно авторизації MainWindow.
wnd_AdminAddWorker	Вікно, де адміністратор системи має змогу зареєструвати нового працівника банку. Можна повернутись до wnd_AdminMain.
wnd_AdminAddBank	Вікно, де адміністратор системи має змогу зареєструвати новий банк в базі даних. Можна повернутись до wnd_AdminMain.
wnd_Worker_Main	Головне вікно працівника банку (менеджера). Тут він підтверджує або відхиляє запити клієнтів на створення рахунку в банку. Сожна повернутись на вікно авторизації MainWindow.

### Висновки до розділу

В даному розділі було описано основні аспекти реалізації розроблюваного продукту.

Наведено реалізацію архітектури додатку шляхом побудови діаграми класів системи. Була описана реалізація та призначення кожного класу предметної області додатку.

Було описано основні алгоритми та послідовності, що були використані для розробки додатку. Вони дали змогу підвищити продуктивність та швидкодію додатку, проте планується й подальша оптимізація розроблених алгоритмів.

Також у розділі було описано реалізацію структури бази даних, наведено повну інформацію про назву та тип даних кожного поля таблиць БД. Наведена

реалізація бази даних є найбільш зручною для редагування та зберігання даних розроблюваної системи.

У розділі також описана схема переходу між вікнами додатку та наведена таблиця з коротким описом усіх вікон та інформацією про вікна, до яких можна перейти з кожного з них.

					ІТ61.050БАК.004 ПЗ	Лист
						50
Змн.	Лист	№ докум.	Підпис	Дата		



## 4 ТЕСТУВАННЯ ПРОГРАМИ

### 4.1 Загальні відомості

При розробці будь-якого програмного забезпечення необхідним є переконатись в його працездатності та правильному функціонуванні. Саме тому аналіз якості та тестування є необхідним аспектом розробки системи для управління банківськими рахунками.

Існує багато методів тестування програмного забезпечення, але проведення ефективних тестів на складних продуктах — це не лише процес дослідження, а й питання створення та суворого дотримання певних жорстких кроків. Одне з визначень тестування: процес опитування продукту для оцінки; "питання" — це те, що тестер намагається зробити з продуктом, а продукт дає відповідь на поведінку сценарію тестера. Хоча інтелектуальний процес більшості тестів — це ні що інше, як огляд та перевірка, слово "тест" означає динамічний аналіз продукту, впевненість, що додаток працює безперебійно. Якість програми може відрізнятися від системи до системи, проте загальновизнані критерії є обов'язковими: надійність, стабільність, портативність, простота обслуговування та практичність.

Основним його завданням є перевірка правильності роботи системи, її швидкодії та стійкості перед великими навантаженнями. Тому проводячи тестування даного проєкту, необхідно звернути увагу на такі аспекти роботи:

- найперше, це правильність та швидкодія здійснення грошових переказів між двома картами;
- правильність записів та редагування реєстраційних даних користувачів;
- правильність визначення строку дії банківської карти;
- швидкодія системи;
- стійкість системи до великого потоку даних.

## 4.2 Модульне тестування

Unit тестування (або ж модульне тестування) — процес в програмуванні, що дозволяє перевірити на коректність окремі модулі вихідного коду програми, набори з одного або більше програмних модулів разом з відповідними керуючими даними, процедурами використання і обробки.

Таке тестування зазвичай використовується для невеличких методів або функцій системи.

Звіт з тестування окремих модулів програмного забезпечення наведений у таблиці 4.1.

Таблиця 4.1 — Тестування окремих модулів програми

Назва методу	Очікуваний результат	Отриманий результат	Статус тесту
ChangePassword	Заміна паролю користувача в базі даних на вказаний новий.	Пароль замінено в базі даних на новий	ПРОЙДЕНО
CheckValidPas	Перевірка правильності введено паролю для даного логіну. Якщо пароль в базі даних не співпадає з вказаним, метод повертає помилку.	Пароль в базі даних не співпадає, метод повернув помилку.	ПРОЙДЕНО

Назва методу	Очікуваний результат	Отриманий результат	Статус тесту
Btn_back_Click	Відкривається вікно інформації про банк, який було обрано в списку банків. Минуле вікно закривається.	Відкривається вікно інформації про банк, який було обрано в списку банків. Минуле вікно закривається.	ПРОЙДЕНО
Btn_card_operations_Click	Відкривається вікно здійснення фінансових операцій за картою, яку було обрано в списку карт клієнта. Вікно карт клієнта закривається.	Відкривається вікно здійснення фінансових операцій за картою, яку було обрано в списку карт клієнта. Вікно карт клієнта закривається.	ПРОЙДЕНО
GetCurrentID	Метод має повертати ідентифікатор користувача, від іменем якого відбувається поточний сеанс.	Метод не повертає ідентифікатор користувача, від іменем якого відбувається поточний сеанс.	ВИПРАВЛЕНО

Назва методу	Очікуваний результат	Отриманий результат	Статус тесту
CardNumberGenerator	Генерація 16значного номеру карти, він має бути унікальним відносно інших номерів карти, записаних в базу даних системи. Метод повертає 16значне число.	Відбулась генерація 16значного номеру карти, він є унікальним відносно інших номерів карти, записаних в базу даних системи. Метод повертає 16значне число.	ПРОЙДЕНО
CardPINGenerator	Генерація 4значного секретного ПІН-коду карти. Метод має повертати 4значне число.	Відбулась генерація 4значного секретного ПІН-коду карти. Метод повернув 4значне число.	ПРОЙДЕНО
AddClientToBD	Система має зчитати дані з відповідних полів інтерфейсу та додати новий запис в таблицю клієнтів, заповнивши відповідні поля введеними даними.	Було зчитано дані з відповідних полів інтерфейсу та додано новий запис в таблицю клієнтів, заповнено відповідні поля введеними даними.	ПРОЙДЕНО

Назва методу	Очікуваний результат	Отриманий результат	Статус тесту
GetBankFromBD	Метод має знайти в таблиці банків в БД запис з заданим ідентифікатором. Метод має повертати об'єкт класу банк.	Метод знайшов в таблиці банків в БД запис з заданим ідентифікатором. Метод повернув об'єкт класу банк.	ПРОЙДЕНО
UpdateCardMoney	Метод має оновити дані про значення коштів на карті за заданим ідентифікатором. Та зробити відповідні записи до бази даних.	Метод оновив дані про значення коштів на карті за заданим ідентифікатором. Та зробив відповідні записи до бази даних.	ПРОЙДЕНО

#### 4.3 Тестування сценаріїв

Тестування сценаріїв роботи програми є важливим етапом перевірки працездатності розроблюваного програмного забезпечення. Доступ до основних функцій має бути не тільки простим та інтуїтивно зрозумілим, а й точним у своїй послідовності. Тестування сценаріїв має дати розуміння правильності роботи сценаріїв роботи з додатком шляхом порівняння запланованого та дійсного сценаріїв роботи користувача для виконання тієї чи іншої дії.

На таблицях 4.2 — 4.4 наведені звіти з тестування сценаріїв виконання основних функцій програмного забезпечення.

Тестування процесу реєстрації є важливим етапом перевірки даного програмного забезпечення, адже через цю ділянку програми проходить майже

кожен користувач системи. Звіт з тестування сценарію реєстрації нового користувача 4.2.

Таблиця 4.2 — Тестування сценарію реєстрації нового користувача

Назва	Реєстрація нового користувача
Опис	Реєстрація надає можливість користуватись системою. Цей етап має пройти кожен клієнт
Очікуваний сценарій	Після запуску програми відкривається вікно авторизації. В ньому треба натиснути кнопку «Реєстрація» та перейти на вікно реєстрації. Там треба ввести свої дані у відповідні поля та натиснути кнопку «Зареєструватись».
Дійсний сценарій	Після запуску програми відкривається вікно авторизації. В ньому було натиснуто кнопку «Реєстрація» після чого відбувся перехід на вікно реєстрації. Там введено вигадані дані у відповідні поля та натиснено кнопку «Зареєструватись». Реєстрація завершена.
Статус	Пройдено

Важливим є також тестування сценарію подачі заяви на реєстрацію нового рахунку, адже це одна із основних функцій системи. Звіт з тестування сценарію реєстрації нового рахунку наведено в таблиці 4.3.

Таблиця 4.3 — Тестування сценарію реєстрації нового рахунку

Назва	Реєстрація нового рахунку
Опис	Користувач має можливість подати заяву на реєстрацію нового рахунку в банку.
Очікуваний сценарій	Після запуску програми відкривається вікно авторизації. Необхідно ввести дані у відповідні поля, після цього відкривається профіль користувача. Після натиснення на

	кнопку «Банки» відкривається вікно з інформацією про банки, в яких користувач зареєстрований. Необхідно обрати в таблиці банк в якому треба створити рахунок та натиснути на кнопку «створити рахунок». Запит буде передано менеджеру.
Дійсний сценарій	Після запуску програми відкривається вікно авторизації. Після введення даних у відповідні поля відкривається профіль користувача. Після натиснення на кнопку «Банки» відкривається вікно з інформацією про банки, в яких користувач зареєстрований. Було обрано в таблиці банк в якому треба створити рахунок та натиснено на кнопку «створити рахунок». Запит буде передано менеджеру.
Статус	Пройдено

Здійснення транзакцій є однією з основних функцій додатку, тому перевірка правильності роботи сценарію роботи з ними є необхідним при тестуванні сценаріїв системи. Звіт з тестування сценарію здійснення транзакції наведено в таблиці 4.4.

Таблиця 4.4 — Тестування сценарію здійснення транзакції

Назва	Здійснення транзакції
Опис	Користувач має можливість здійснювати грошові перекази з обраної карти на будь-яку карту зареєстрованого в системі банку.
Очікуваний сценарій	Після запуску програми відкривається вікно авторизації. Необхідно ввести дані у відповідні поля, після цього відкривається профіль користувача. В таблиці з картами користувача необхідно обрати карту з якої буде

	здійснюватися переказ та натиснути на кнопку «транзакції». У вікні транзакцій необхідно ввести секретний номер даної карти, номер карти отримувача та суму коштів, що необхідно переказати. Після цього з'явилося повідомлення про успішне завершення транзакції.
Дійсний сценарій	Після запуску програми відкрилося вікно авторизації, було введено дані у відповідні поля, після цього відкрився профіль користувача. В таблиці з картами користувача було обрано карту, з якої буде здійснюватися переказ та натиснено на кнопку «транзакції». У вікні транзакцій було введено секретний номер даної карти, номер карти отримувача та суму коштів, що необхідно переказати. Після цього з'явилося повідомлення про успішне завершення транзакції.
Статус	Пройдено

#### 4.4 Тестування навантаженням

Подібна система може мати великий потік даних, тому треба перевірити наскільки вона зможе з ним справитись за допомогою тестування навантаженням. Код програми було модифіковано таким чином, що програма автоматично буде генерувати різну кількість запитів або дій, щоб перевірити як буде справлятися дана система з таким навантаженням. Звіт з тестування навантаженням процес реєстрації користувачів наведено в таблиці 4.5.

Така велика затримка виникла, скоріш за все, на рівні обробки запиту самою програмою, адже при розробці проєкту було обрано достатньо потужну базу даних, затримки від якої при такій кількості дій не очікується.



Таблиця 4.5 — Тестування навантаженням процес реєстрації користувачів

Назва	Реєстрація нового користувача
Опис	Процес реєстрації клієнта є дуже важливим об'єктом тестування, адже він є першим етапом знайомства користувача з системою.
Запит на реєстрацію 1 000 користувачів одночасно	Відчутних змін не відбулося. Система продовжила свою роботу без помітної затримки.
Запит на реєстрацію 10 000 користувачів одночасно	Система обробила запити та продовжила свою роботу з помітною затримкою.
Запит на реєстрацію 100 000 користувачів одночасно	Система не змогла обробити запити за короткий проміжок часу та призупинила свою роботу на період, що є неприйнятним для користувача.

## Висновки до розділу

Тестування допомогло виявити деякі недопрацювання дало змогу зрозуміти вектор подальших дій над програмою. Система не витримала сильного навантаження, тому планується подальша оптимізація коду програми.

Наразі тестування основного функціоналу завершено. У подальшому планується тестування інших компонентів програми.

## 5 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для більш успішного впровадження програмного забезпечення необхідно створити зрозумілу та поетапну інструкцію використання. Це дозволить користувачам безперешкодно знаходити необхідні їм функції та дасть відповіді на деякі запитання, що виникнуть під час користування програмою.

### 5.1 Підготовка до запуску програми

Розроблене програмне забезпечення використовує деякі сторонні програми, без яких проєкт не зможе повноцінно працювати. При впровадженні програмного забезпечення необхідним є перевірити та вказати основні етапи підготовки до його запуску.

#### 5.1.1 Вимоги до програмного та апаратного забезпечення

Для використання даної системи для управління рахунками необхідно мати наступні системні та апаратні характеристики:

- RAM мінімум 1 ГБ;
- OS Windows 7 та новіші;
- встановлений .NET Framework;
- 2-10 ГБ фізичної пам'яті (в залежності від запланованого навантаження системи інформацією;
- частота процесора 1 ГГц і вище.

#### 5.1.2 Вимоги для запуску

Для повноцінної роботи з базою даних та сервером потрібно мати встановлене таке програмне забезпечення:

					ІТ61.050БАК.004 ПЗ	Лист
						60
Змн.	Лист	№ докум.	Підпис	Дата		

- пакет LINQ TO SQL;
- СУБД MS SQL Server;
- Microsoft Visual Studio 2013 і новіше для запуску проєкту.

### 5.1.3 Підключення до MS SQL Server

Основою роботи такої системи є зберігання інформації у базі даних, тому встановити підключення до БД є одним з найважливіших етапів підготовки до запуску проєкту.

Для роботи з базою даних необхідно підключити MS SQL Server до проєкту. Інструкція з підключення:

1. Відкрити файл App.config, який знаходиться у кореневій папці проєкту.
2. Змінити значення атрибутів name та connectionString у відкритому файлі конфігурації App.config (так ми зможемо підключитись до БД).

Це необхідно для збереження історії транзакцій та інформації про зареєстрованих клієнтів банку, працівників та реєстрацію самих банків.

### 5.2 Інструкція з використання клієнту банку

При запуску проєкту, на екрані з'являється вікно авторизації (рис. 5.1), де ви маєте ввести свій зареєстрований логін та пароль та натиснути кнопку «Вхід».

При введенні неправильних даних система повідомить про помилку (рис. 5.2). Це може означати, що ви ввели невірні дані, або ще не зареєструвались у системі.

Якщо ви новий користувач, необхідно натиснути кнопку «Реєстрація», яка переведе вас на вікно реєстрації нового користувача системи (рис. 5.3).

## Вхід до інформаційної системи банків

Логін

---

---

Пароль

---

---

Увійти

---

Реєстрація

---

Рисунок 5.1 — Авторизація до системи

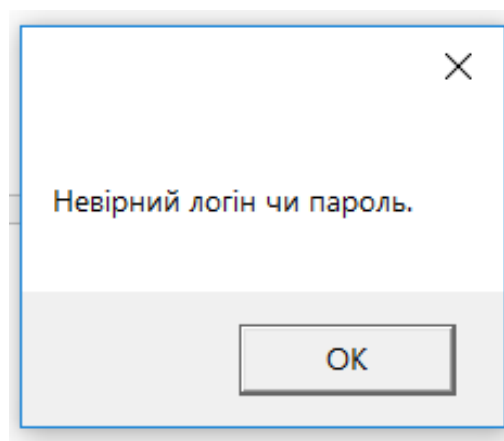


Рисунок 5.2 — Помилка авторизації до системи

## Реєстрація

Прізвище	Ім'я
<input type="text"/>	<input type="text"/>
Логін для входу	
<input type="text"/>	
Пароль	Повторіть пароль
<input type="password"/>	<input type="password"/>
<input type="button" value="Зареєструватись"/>	
<input type="button" value="Назад"/>	

Рисунок 5.3 — Реєстрація нового клієнта банку в системі

Для реєстрації необхідно ввести прізвище, ім'я, логін та пароль у відповідних підписаних полях. Після чого натиснути кнопку «Зареєструватись». Програма перевірить правильність введених даних. Якщо дані було введено невірно, система повідомить про помилку, після чого треба перевірити правильність усіх введених даних.

Після успішної реєстрації ви потрапите до свого профілю (рис. 5.4).

В профілі ви можете бачити список своїх банківських рахунків. Справа від списку розташована кнопка «+». Якщо ви бажаєте створити новий банківський рахунок, необхідно натиснути на неї, після чого відкриється вікно реєстрації нового рахунку (рис. 5.5).

Олександр Бондаренко

Змінити пароль

Мої рахунки

9189 2047 6404 9119 (First Bank) - \$60

0739 6168 6278 7495 (Second Bank) - \$0

+

Дії з картою

Змінити акаунт

Рисунок 5.4 — Профіль клієнта банку.

Існуючі банк

Second Bank (0664544213)

First Bank (0664544212)

Створити карту для обраного банку

Назад

Рисунок 5.5 — Вікно реєстрації нової банківської карти

У цьому вікні необхідно обрати банк зі списку зареєстрованих в системі та натиснути кнопку «Зареєструвати рахунок у цьому банку».

Якщо ж ви вже маєте власні рахунки та бажаєте здійснити з ними певні транзакції, у профілі необхідно обрати карту для взаємодії зі списку, після чого натиснути на кнопку «Дії з картою», що переведе вас у вікно транзакцій (рис. 5.6).

**Транзакції**

Оберіть операцію

Поповнити рахунок ▾

Сумма (\$)

PIN

Підтвердити

Назад

Історія транзакцій

Поповнення рахунку на суму \$120 28.12.2018 5:39:10  
 Зняття з рахунку \$30 28.12.2018 5:48:31

Оновити

Рисунок 5.6 — Вікно транзакцій

Зі списку зліва зверху потрібно обрати операцію, котру ви бажаєте провести з картою. Для цього потрібно натиснути на стрілочку вниз, щоб побачити меню вибору операції (рис. 5.7).

Оберіть операцію

▾

Поповнити рахунок

Зняти готівку

Переказ на іншу карту

Рисунок 5.7 — Меню вибору транзакції

Після вибору транзакції з'являється певна кількість полів для вводу. Для виконання транзакції необхідно ввести необхідні дані та натиснути кнопку «Підтвердити».

В правій частині вікна ви можете побачити запис історії транзакцій. Для перевірки результатів щойно виконаної транзакції необхідно натиснути на кнопку «Оновити» і прочитати відповідне поле.

### 5.3 Інструкція з використання адміністратора

Після авторизації у систему в ролі адміністратора інформаційної системи, відкривається вікно профілю адміністратора.

На екрані ви можете бачити список зареєстрованих в системі банків, а саме їх назву та контактний номер телефону, кнопки «+», «Зміна даних про банк», «Реєстрація працівника», та «Назад» (рис. 5.8).

The screenshot displays the 'ADMIN' interface for the 'Банки системи' (System Banks). It features a list of banks with the following entries: 'Second Bank (0664544213)' and 'First Bank (0664544212)'. To the right of the list is a blue button with a white '+' symbol. Below the list are three buttons: a blue button labeled 'Зміна даних про банк' (Change bank data), a grey button labeled 'Реєстрація працівника' (Employee registration), and a white button labeled 'Назад' (Back) at the bottom.

Рисунок 5.8 — Форма профілю адміністратора інформаційної системи

Після натиснення кнопки «+», що розташована в правій частині вікна, відкривається вікно форми реєстрації нового банку (рис. 5.9). На екрані присутні поля для вводу тексту з надписами «Назва банку», «Адреса банку», «Телефон», потрібно ввести необхідну інформацію у відповідні поля. Після цього необхідно переглянути введену інформацію та натиснути кнопку «Додати банк до системи».



### Реєстрація банку

Назва банку

Адреса банку

Телефон

Додати банк до системи

Назад

Рисунок 5.9 — Форма реєстрації нового банку

Якщо введені дані не є валідними, система повідомить про помилку при реєстрації нового банку (рис. 5.10).

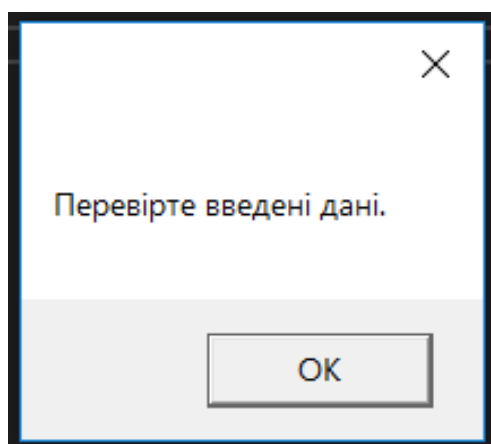


Рисунок 5.10 — Помилка при реєстрації нового банку

Якщо введені дані є валідними, новий банк буде зареєстровано, після чого відкриється вікно профілю адміністратора інформаційної системи.

Також при натисненні кнопки «Назад» можна повернутись до форми профілю адміністратора інформаційної системи.

При натисненні на кнопку «Зміна даних про банк» відкриється вікно зміни реєстраційних даних для обраного в списку банків банку, проте наразі дана функція знаходиться на стадії розробки.

При натисненні кнопки «Назад», ви перейдете на вікно авторизації до інформаційної системи банків.

Для реєстрації нового працівника необхідно обрати банк, для якого буде зареєстровано нового працівника, та натиснути на кнопку «Реєстрація працівника». Якщо при натисненні кнопки, не буде обрано жодного банку зі списку, система повідомить про помилку при запиті на реєстрацію нового клієнта (рис. 5.11).

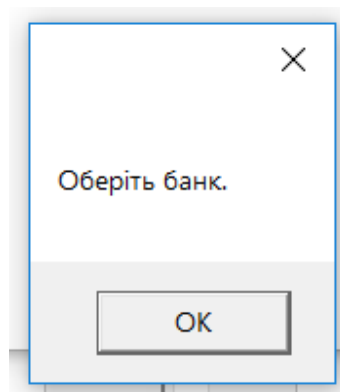


Рисунок 5.11 — Помилка при спробі реєстрації нового працівника для банку

Після успішної обробки запиту на реєстрацію нового працівника відкриється вікно з формою реєстрації нового працівника (рис. 5.12).

На вікні з формою для реєстрації нового працівника банку присутні поля для вводу даних з відповідними назвами «Прізвище», «Ім'я», «E-mail», «Логін для входу», «Пароль», «Повторіть пароль». Для реєстрації необхідно ввести усі необхідні дані у відповідні поля та натиснути кнопку «Зареєструвати».

При натисненні кнопки, система перевірить валідність введених даних. Якщо введені дані не будуть задовольняти потреби, система повідомить про помилку.

Якщо введені дані є валідними, нового працівника банку буде зареєстровано, після чого відкриється вікно профілю адміністратора інформаційної системи.

Реєстрація працівника банку

Прізвище	Ім'я
<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>
E-mail	
<input style="width: 90%;" type="text"/>	
Логін для входу	
<input style="width: 90%;" type="text"/>	
Пароль	Повторіть пароль
<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>

[Назад](#)

Рисунок 5.12 — Форма реєстрації нового працівника банку

Також при натисненні кнопки «Назад» можна повернутись до форми профілю адміністратора інформаційної системи.

### Висновки до розділу

Було описано необхідні вимоги та кроки для встановлення системи. Також було описано основні дії користувачів та адміністратора для полегшення взаємодії з системою.

Описані інструкції сприятимуть покращенню впровадження розроблюваного програмного забезпечення.

## ВИСНОВКИ

Результатом розробки є внутрішньобанківська інформаційна система для управління рахунками банків. Було створено десктоп-додаток для операційної системи Windows. Розроблена система дає змогу реєструвати нових клієнтів та працівників банку, створювати нові банківські рахунки в системі, редагувати дані та видаляти інформацію. Також система підтримує здійснення різних фінансових операцій та перегляду історії транзакцій. Для забезпечення приватності даних було розроблено авторизацію, яка визначає рівень доступу користувача системи. Застосунок має три рівні доступу — клієнт банку, працівник та адміністратор.

Перед розробкою даного продукту було вивчено ринок банківських додатків, проаналізовано різні дослідження у цій предметній області та проведено власне. Це допомогло підтвердити актуальність створення такого додатку та визначити основний функціонал системи відповідно до потреб споживачів.

Для реалізації інтерфейсу проєкту була обрана платформа WPF, в основі якої лежить високорівнева мова програмування C#. Для забезпечення збереження усіх даних інформаційної системи було використано СУБД MS SQL Server.

В ході виконання дипломного проєкту було побудовано десктоп додаток, що має зручний та приємний інтерфейс, високий рівень швидкодії та надійне сховище інформації. Дана інформаційна система потребує раду оновлень для приведення її до досконалого вигляду. Планується подальша робота над проєктом та його вдосконаленням. Для розвитку даної інформаційної системи банків необхідно додати ряд нових функцій, а саме зміна реєстраційних даних банку та ряд можливостей для працівників банку. Також можливе покращення графічного інтерфейсу, підвищення його зручності та лаконічності. Також потрібно розробити можливість перегляду доступних банків на карті світу з відображенням інформації про них.

					IT61.050БАК.004 ПЗ	Лист
						70
Змн.	Лист	№ докум.	Підпис	Дата		

В проєкті описано процес розробки програмного забезпечення, вибір основних методів та інструментів для реалізації даного продукту. Також було описано розробку та реалізації бази даних для збереження інформації системи.

Дана система була протестована на правильність виконання окремих функцій, правильність роботи та швидкодію при навантаженнях. Планується подальше тестування програми.

Для впровадження та супроводу програмного забезпечення було розроблено детальну інструкцію з підготовки до встановлення, встановлення та користування програмним забезпеченням.

					ІТ61.050БАК.004 ПЗ	Лист
						71
Змн.	Лист	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Банк. URL: [https://old.bank.gov.ua/control/uk/publish/article?art\\_id=123122&cat\\_id=122978](https://old.bank.gov.ua/control/uk/publish/article?art_id=123122&cat_id=122978)
2. Електронний банкінг. URL: <https://pidru4niki.com/17910211/finansi/>
3. Інтерфейс користувача. URL: [https://uk.wikipedia.org/wiki/Інтерфейс\\_користувача](https://uk.wikipedia.org/wiki/Інтерфейс_користувача).
4. Що таке база даних?. URL: <http://apeps.kpi.ua/shco-take-basa-danykh>
5. Різниця між компілятором, редактором та IDE . URL: <https://presmarymethuen.org/uk/dictionary/what-is-the-difference-between-a-compiler-an-editor-and-an-ide-can-you-explain-in-laymen-s-terms/>
6. Бази даних і СУБД. URL: <https://timeweb.com/ru/community/articles/bazy-dannyh-i-subd-1>
7. WPF . URL: <https://metanit.com/sharp/wpf/1.php>
8. Рефакторинг. URL: <https://refactoring.guru/ru/refactoring>
9. Уніфікація (техніка). URL: [https://uk.wikipedia.org/wiki/Уніфікація\\_\(техніка\)](https://uk.wikipedia.org/wiki/Уніфікація_(техніка))
10. Що таке фреймворк?. URL: <https://saikt-online.ru/chto-takoe-frejmworki-i-dlya-chego-oni-nuzhny/>
11. Дослідження Mastercard. URL: <https://newsroom.mastercard.com/eu/uk/press-releases/digitalization-trust-survey/>
12. Користування цифровими додатками в Україні, 2019. URL: [https://newsroom.mastercard.com/eu/files/2019/06/Digitalization\\_infographics\\_UA.jpg](https://newsroom.mastercard.com/eu/files/2019/06/Digitalization_infographics_UA.jpg)
13. В яких банків немає додатків. URL: <https://minfin.com.ua/ua/2019/10/18/39314319/>
14. Мої рахунки. Приватбанк. URL: <https://privat24.privatbank.ua/p24/web/#statements/cards/>
15. Головна. Альфа-Банк. URL: <https://auth.alfabank.com>

					IT61.050БАК.004 ПЗ	Лист
Змн.	Лист	№ докум.	Підпис	Дата		72